*Digital Crime Analyser*

Team 29

Branson Tian ([2101848@sit.singaporetech.edu.sg](mailto:2101848@sit.singaporetech.edu.sg)),

Shathiya Sulthana ([2102605@sit.singaporetech.edu.sg](mailto:2102605@sit.singaporetech.edu.sg)),

Fitri Raihan ([2101130@sit.singaporetech.edu.sg](mailto:2101130@sit.singaporetech.edu.sg)),

Marhakim ([2102932@sit.singaporetech.edu.sg](mailto:2102932@sit.singaporetech.edu.sg)),

Kenneth Chee ([2100788@sit.singaporetech.edu.sg](mailto:2100788@sit.singaporetech.edu.sg))

*Abstract*— *A team of undergraduates from Singapore Institute of Technology, developing a tool to help crime investigators better investigate digital crimes extracting datasets, and predict the type of attacks that occurred based on datasets collected*

Keywords— Digital Crime Analyzer, Dataset Analysing, Machine Learning, Attack Prediction, Tensorflow-Keras

# Introduction (*Heading 1*)

Digital crimes are becoming increasingly popular. These are criminal activities that target computers and networks or use a computer as a tool for these malicious activities. There are many forms of Digital Crimes, such as FTP BruteForce, DOS, DDOS, and many more, depending on the criminal’s intent to crack a system or steal data.

In this project, the team’s task is to analyze datasets and extract them into a program that further supports criminal investigations. Using a Graphical User Interface (GUI), the user includes the dataset via the input and the dataset display showing a search and export function. The dataset is extracted for analysis and displays the output to the user, and a statistical graph shows the prediction of the type of attack using machine learning.

# Method

## Selection of data

For the dataset selection, we decided to use the dataset provided in our problem statement, https://www.unb.ca/cic/datasets/ids-2018.html. This dataset includes seven different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside. It has numerous information about the event logs and network traffic from multiple machines and servers. The dataset also included detailed descriptions of intrusions and protocols [1]. To get the dataset, we installed the AWS CLI, a command-line interface that manages AWS services. After installing, we had to run this command “aws s3 sync --no-sign-request --region <your-region> "s3://cse-cic-ids2018/" dest-dir”. The command <your-region> lets us choose from the list of regions that AWS has provided us [1]. By running the above-stated command, three kinds of files will be available: PCAP, Windows Event Logs, and CSV files of data extracted with CICFlowMeter-V3. We are using the data of the CSV files to analyze them with our program.

## Accessing Dataset

To allow our program to analyze the dataset, we will need to open the dataset to read its content. We are using Tkinter Py module to enable us to spread the dataset, but it is also a GUI toolkit, which is a great foundation for achieving our desired result. By using the file dialog function, will allow us to assign our chosen file path. Fig. 2a shows how we defined a function for the file dialog.

A screenshot of a computer

Description automatically generated with medium confidence

Fig.2a Filedialog implementation

## Reading Dataset

Once we access the dataset, we need our program to read the contents to realize the output. We use Pandas to gather all the data through the CSV file, which passes to clean and process the data.

Pandas have a list of features with data structure, as shown below [2]

Pandas

* Uses “DataFrame” to create the list (2D)
* Groups data together and reshapes for easier visualization
* Looking through the data to provide statistics

Fig.2b shows the implementation of Pandas in our program.

Text

Description automatically generated

Fig.2b Pandas read function

To ensure the user chooses the correct file format for pandas to read, we incorporated the except function to prompt the user to select an invalid file.

## Machine Learning

The team’s goal is for the program to provide the user with a prediction on the type of attack that might have occurred based on the dataset. The prediction model is created using machine learning, and it predicts the type of attack based on the dataset. The team chose Tensorflow-Keras to assist us in creating the prediction model as it uses a deep-learning model to make predictions such as classification and regression predictive modeling. We created a set of data with labels that we utilized to train the prediction model. The data will be turned into a tensor object, making it easier for Keras to access it. We used a multi-layer perceptron where the first layer mainly states the input size, and the last layer will have a unit size of 13 which will be the number of attacks we will be accessing. In compiling, we used “Adam” for the optimizer and “sparse\_categorical\_crossentropy” for the loss. Then we trained the sample data for 600 epochs with a batch size of 75. [3]. Fig.2c shows our implementation of Tensorflow-Keras in our program.

Text

Description automatically generated

Fig.2c Machine Learning implementation

## User Perspective

We have created a GUI that the user can interact with for the functions to be executed. We are using Tkinter GUI toolkit as it has easy implementation and is simple for users to understand. The first tab of the GUI (Data uploaded) has the following functions:

* Display Dataset
* Button to browse for the file
* Button to load the file
* Input boxes for users to indicate the values to search in the dataset
* Button to execute the search

The GUI also has another tab (Processed data), which displays:

* Prediction output
* Buttons to display data in graphs
* Button to export file

Fig.2d and Fig.2e show the result of Tkinter GUI implementation.

**Graphical user interface

Description automatically generated**

Fig.2d Tkinter GUI “Data Uploaded” Tab

**Graphical user interface, application, Word

Description automatically generated**

Fig.2e Tkinter GUI “Processed Data” Tab

## Statistic

# With the data collected from the CSV, we can create statistical data to show to the user. We are using Matplotlib as it contains data visualization and graphical plotting library to help with making the following :

* Bar Graph
* Pie Chart
* Donut Chart

We start off by converting only the destination ports into a list. During the conversion, the program will also sort it out from highest to lowest value. From there, only take the top 5 values by making the range of values end at the 5th index. With the values in the list, using Matplotlib, we converted it into a bar graph. Its bar graph will indicate the port number and a range of numbers whereby the number of times it was attacked at the side. In terms of the bar graph, we will find the top five destination ports that have been attacked and display the results in a bar graph.

The pie chart displays the dataset’s comprehensive data, the different types of attacks, and the number of times the attack has occurred. A legend is displayed at the side to reflect the data representation of its color in the pie chart.

Lastly, for the donut chart, we will find the top 5 attacks which occurred. Generating the donut chart will be the same as the pie chart, and getting the data will be the same as our bar graph. A legend will be provided on the side to reflect the donut chart’s data representation of its color.

## Export

Users will export the file when the program has displayed the prediction in the ‘Processed Data’ tab. The exported file will be in a CSV format containing rows of values like prediction, protocol, destination port, and timestamp. Apart from that, when the user selects statistical data such as either a bar graph, pie chart, or donut chart, the user will be able to save and export out the image of the charts with the help of the Matplotlib navigation toolbar.

# result and insight

The figures below show an example of a dataset that has been loaded to the program, as well as the output result.

**Graphical user interface

Description automatically generated**

Fig.3a Digital Crime Analyzer with dataset loaded

## Process Uploaded Tab

Based on the result, it shows the dataset which has been filtered for the user to view, and by inserting a value in the search input box, it will only display the relevant data as shown in fig.3b below,

**Graphical user interface

Description automatically generated**

Fig.3b Search Function

## Processed Data Tab

This will display the prediction output as shown in Fig.3c below,

**Graphical user interface, table

Description automatically generated**

Fig.3c Prediction output

By clicking on the respective graph buttons, the findings will be displayed in the form of the three graphs that have been implemented, as shown in Fig.3d, 3e, and 3f below.

**Chart, pie chart

Description automatically generated**

Fig.3d Donut Chart

**Chart, pie chart

Description automatically generated**

Fig.3e Pie Chart

**Chart, bar chart

Description automatically generated**

Fig.3f Bar Graph

The export button will allow the user to export the table dataset into their local machine, as shown in fig.3g below.

**Table

Description automatically generated with low confidence**

Fig.3h Exported File

## Insight

The investigators will better determine what type of attack is more often based on the outputs. With the data visualization, the users can better look at and understand what cyber-attacks there are. The bar graph depicts which destination ports are most frequently attacked, while the pie chart illustrates all of the different types of attacks that could occur, and the donut chart displays the top five attacks that are most likely to occur. The investigators can narrow down the specifics of the data required using the filtered data from the search query. The exported results simplify the data, making it easier for users to interpret and distribute the information. Machine learning can distinguish between different sorts of attacks and their properties, allowing the user to comprehend the course of each attack and prepare for future occurrences.

# Conclusion

This project’s task is to help crime investigators to view previous attacks and determine the attack that has occurred. This can be taken a step further by predicting an attack while it is happening. It might also help by providing suggestions to prevent or block such attacks. Knowing which attack takes place the most can help investigators prepare more towards guarding against it. It also showcases visualization of data which can be helpful as curating data into a form that is easier to understand, highlighting the trends and outliers. This project let us learn about machine learning, designing a quick UI interface for users, and utilizing Python as a language. For future works, if there is a need to use Python again, the knowledge we have gained can be used and be implemented more efficiently than before. Dealing with data can also come in handy as future projects might utilize databases and data analytics. Knowing more about diverse libraries will be helpful in the future, as many projects with the same goal will have comparable libraries.

Tasks allocated among each of us are split into 5 essential modules:

* Shathiya – Main program control, machine learning (Tensorflow-Keras), and report writing
* Kenneth - Machine learning (Tensorflow-Keras) and report writing
* Marhakim – File Handler, search and report writing
* Raihan – Graph plotting (Matplotlib) and report writing
* Branson – GUI (Tkinter) and report writing

# Acknowledgment

We like to thank Professor Daniel Wang Zhengkui for providing us with the resources we needed to start the project. Professor Arthur Loo Wee Yeong deserves special gratitute for advising us on the challenges we encountered when incorporating machine learning into our program.

# References

1. "IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB", Unb.ca. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html
2. "User Guide — pandas 1.3.3 documentation", Pandas.pydata.org, 2021. [Online]. Available: https://pandas.pydata.org/docs/user\_guide/index.html.
3. "Keras Tutorial: Deep Learning in Python", *https://www.datacamp.com*, 2021. [Online]. Available: https://www.datacamp.com/community/tutorials/deep-learning-python?utm\_campaignid=282657555&utm\_adgroupid=1147890695096458&utm\_device=c&utm\_keyword=keras&utm\_matchtype=p&utm\_network=o&utm\_adpostion=&utm\_creative=&utm\_targetid=kwd-71743615746804:loc-164&utm\_loc\_interest\_ms=&utm\_loc\_physical\_ms=164&msclkid=ec235b1bc69d1202591b3a6af3603710&utm\_source=bing&utm\_medium=cpc&utm\_campaign=NEW%20Granular%20Topics%20(via%20DSA%20insights)%20%7C%20Tier%201%20BING&utm\_term=keras&utm\_content=community%2Ftutorials%2Fdeep-learning-python.
4. “Keras with TensorFlow Course - Python Deep Learning and Neural Networks for Beginners Tutorial,” *Youtube.com*, 2021. [Online]. Available: https://www.youtube.com/watch?v=qFJeN9V1ZsI.