CSCE5214\_Sec1\_Assignment – 1

**Task-1**

Step-1:

Downloaded the files from the link.

Step-2:

# Three log files were processed by a Python software that also created three tables, each of which was tied to a different situation. These circumstances include injecting RPM, injecting fuel flow, and not injecting at all. The last element in the collection serves as an indicator of the information in the tables regarding the CAN injection state. The value is set to 1 if an injection or attack has occurred, and 0 otherwise.

# Task-2

Step-1:

Graphical user interface

Description automatically generated

Graphical user interface, chart, histogram

Description automatically generated

Graphical user interface, chart, histogram

Description automatically generated

In the first row of figures, three different scenarios—fuel flow as speed injection, RPM injection, and no injection—are used to show how speed changes over time. The plots with injection reveal a highly consistent and dependable data pattern, in contrast to the plot without injection.

The three identical situations' RPM evolution is depicted in the second row of charts. In contrast to the plot without injection, the plot with injection displays a high, stable, and regular data pattern. Also, compared to the plot with FFF injection, the RPM values in the plot with RPM injection fluctuate less frequently.

The correlation between speed and RPM for each scenario is shown in the third row of charts. RPM and speed appear to be positively correlated in all three charts. Compared to the plots without injection, which are more dispersed and unorganized, the plots with injection show a more tightly packed and consistent data pattern.

In conclusion, adding data to the log files produces a more reliable and coherent data pattern, and the system's behavior can be better understood and its functionality can be increased by taking advantage of the positive linear relationship between speed and RPM.

Step-2:

Graphical user interface, chart, waterfall chart

Description automatically generated

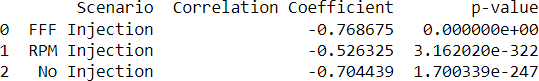
# Observations:

These frequency graphs show the speed and RPM data distribution for three separate scenarios. The frequency plot for speed in the fuel flow as speed injection (FFF) scenario exhibits a bimodal distribution with two peaks, signifying the existence of two separate sets of speed values. The RPM frequency plot, on the other hand, shows a unimodal distribution with just one peak, suggesting that RPM measurements are relatively similar.

The RPM frequency plot in the RPM injection scenario exhibits two peaks, indicating that there are two distinct groups of RPM values. The speed frequency plot, on the other hand, only has one peak, which suggests that speed values are dispersed more evenly.In the no injection scenario, both the speed and RPM frequency plots display a unimodal distribution with 1 peak, indicating a more uniform data distribution than in the injection scenarios. However, message injection may help record a wider variety of system actions.

The frequency plots make us understand how the data is spread out in each scenario.

Step-3:



**Observations:**

According to the analysis, although the association's strength changes between the three scenarios, there is a negative correlation between the two variables.

The fuel flow as speed injection (FFF) scenario exhibits a high negative connection, which means that as speed rises, RPM falls. The scenario without injection also exhibits this same substantial negative connection. The moderate negative correlation in the RPM injection scenario shows that the RPM lowers as the speed rises, but not as dramatically as in the other cases.

The study of the table indicates that the signals have an impact on the operation of the system and the patterns of the data, which may have an impact on how we interpret and evaluate the data.

# Task-3

The result is :

Table

Description automatically generated

# Task-4

The following result is obtained, when we the data is trained using Decision Tree model:

Table

Description automatically generated

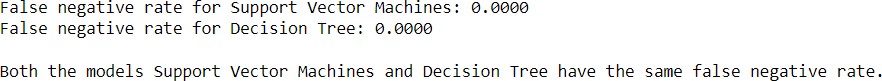
SVM model:

# To summarize, the first paragraph describes the performance metrics of a machine learning model that accurately identified 83% of the data, recognized positive samples with 77% precision, and perfectly identified all samples with a recall value of 1.00. The trade-off between recall and accuracy is also discussed.

# The second paragraph specifically discusses the performance of a Decision Tree model, which achieved a perfect accuracy, precision, and recall values of 1.00. This demonstrates its ability to learn the correlation between the characteristics and the output and classify every sample correctly.

# Furthermore, the Decision Tree model outperformed the SVM model in terms of accuracy, precision, and recall, but additional research may be necessary to ensure there are no biases or issues present.

# Task-5



# There are no missed positive cases and no erroneous negative predictions based on each model's perfect operation. The confusion matrix confirms that neither model is producing any false negatives and is operating as intended. It is expected that both models would have the same false negative rate because they are both operating perfectly.

# Task-6

The investigation of utilizing supervised machine learning to find in-vehicle networks cyberattacks identified a major problem: the scarcity of labeled data. Due to the lack of labeled data, the process is time-consuming and difficult. Consequently, anomalies in the data can be found using unsupervised or semi-supervised machine learning methods.

While fewer attack cases occur in the dataset than normal message instances, the uneven distribution of classes presents additional barrier for supervised machine learning models. This variation may have an adverse effect on the model's accuracy and produce skewed findings. We can solve this problem by balancing the classes using methods like under-sampling, over-sampling, or creating synthetic examples.

There are several ways we can improve the model's accuracy. Using cutting-edge methods like CNNs and RNNs, which can recognize trends and relationships in data over time, is one possibility. The most crucial aspects in the data can also be chosen or engineered, which helps expedite training and improve performance. To improve the security of in-vehicle networks, we can also use anomaly and intrusion detection technology.