|  |  |  |
| --- | --- | --- |
| CSIS 486 | Team One  Riley Dorough  Kayla Echols  Julia Wilkins  Brett Wolff | AI/ML CTI Product Demonstration Assignment – Version Two |
|  | **March 26th, 2022** | **Phase 2 Step 5** |

**Title [Kayla Echols]**

* The goal of this lab is for the students to choose and add three more IT data features to the existing pipeline.

**Abstract [Kayla Echols]**

The team was tasked with adding IT data processing within the Knime environment. Each of the team members focused on a different IT pipeline data input. We learned how auto-binning can be useful and how that can tie into the IT data. We added the following data from the IT csv file: processor performance, interrupts per second, user time, and clock interrupts per second. These data inputs each have a visual to show the data points from the csv file.

**Introduction** [Kayla Echols]

The team was able to connect the IT data into a pipeline and then into the auto-binner which then displays the various visuals. The IT data is different than the OT data because the data is harder to classify. The main pipeline for the IT data includes the nodes CSV reader, the column rename, string manipulation, string to date & time, and column filter. We then used the auto-binner to find the common data points in order to show those points with the visuals. These visuals include both pie charts, value counters, and scatter plots. This lab required the students to have access to Liberty’s VPN and access to the Knime software to complete the lab.

**Method [Riley Dorough]**

Diagram

Description automatically generated

* Pipelines 8, 9, 10, and 11 are offshoots of the data formatting modules initialized in the last cycle of development

**Pipeline 8 – Processor Performance**

* First the column rename module formats the column’s name to be understandable and easy to recognize
* The String manipulation module reformats the time column so that it can be recognized by the datatype conversion module
* The String to Date & Time module formats the timestamp to a continuous Date&Time variable range
* The Column filter cuts out the single processor information to make the resulting table quicker and easier to operate and process
* The Auto Biner dynamically assigns values to each input column based on distribution of data from each column {Processor Performance, Interrupts/ Second, and User Time} these are separated into three bins, flagging far extreme values
* The value counter node lists the number of values in each bin and any null/missing values in a table format for additional data utilization
* Lastly, the Pie Chart graph shows all the binned values in an easy-to-understand format

Chart, pie chart

Description automatically generated

Table

Description automatically generated

**Pipeline 9 – Interrupts Per Second**

* First the column rename module formats the column’s name to be understandable and easy to recognize
* The String manipulation module reformats the time column so that it can be recognized by the datatype conversion module
* The String to Date & Time module formats the timestamp to a continuous Date&Time variable range
* The Column filter cuts out the single processor information to make the resulting table quicker and easier to operate and process
* The Auto Biner dynamically assigns values to each input column based on distribution of data from each column {Processor Performance, Interrupts/ Second, and User Time} these are separated into three bins, flagging far extreme values
* The value counter node lists the number of values in each bin and any null/missing values in a table format for additional data utilization
* Lastly, the Pie Chart graph shows all the binned values in an easy-to-understand format

Chart, pie chart

Description automatically generated

Table

Description automatically generated

**Pipeline 10 – User Time**

* First the column rename module formats the column’s name to be understandable and easy to recognize
* The String manipulation module reformats the time column so that it can be recognized by the datatype conversion module
* The String to Date & Time module formats the timestamp to a continuous Date&Time variable range
* The Column filter cuts out the single processor information to make the resulting table quicker and easier to operate and process
* The Auto Biner dynamically assigns values to each input column based on distribution of data from each column {Processor Performance, Interrupts/ Second, and User Time} these are separated into three bins, flagging far extreme values
* The value counter node lists the number of values in each bin and any null/missing values in a table format for additional data utilization
* Lastly, the Pie Chart graph shows all the binned values in an easy-to-understand format

Chart, pie chart

Description automatically generated

Table

Description automatically generated

**Pipeline 11 – Clock Interrupts Per Second**

* First the column rename module formats the column’s name to be understandable and easy to recognize
* The String manipulation module reformats the time column so that it can be recognized by the datatype conversion module
* The String to Date & Time module formats the timestamp to a continuous Date&Time variable range
* The Column filter cuts out the single processor information to make the resulting table quicker and easier to operate and process
* The Auto Biner dynamically assigns values to each input column based on distribution of data for Clock Interrupts / Second these are separated into three bins, flagging far extreme values
* The bins are colored with the color manager module
* The scatter plot module displays the binned values

Graphical user interface, application, table, Excel

Description automatically generated

**Results [Julia Wilkins]**

The goal for this deliverable was to add three more IT data features to the existing IT pipeline. This presented some new challenges when trying to classify the data. With OT data points such a temperature we could easily set boundaries as to what we considered normal temperature fluctuation. However, classifying IT data was far less intuitive when it came to establishing normal boundaries. So, I decided to an autobinner that bins data based on width. Width binning works by setting the width of the range of each bin. So, data that is very similar in value goes into the same bin. By comparing the ratio of how many values are in each bin, it was found that in most cases at minimum 98% of our data was input into the first bin. We can then say that data in bin 1 is considered normal. Data in bins 2 and 3 is considered abnormal. If we were to see more values being input into bins 2 and 3 then we could safely assume a significant change has been made to the system and could be an indication of something malicious. This premise will be important to further refine classification of IT data types.

As of now we have the following data inputs:

* OT:
  + Pos State
  + Freezer Door State
  + Photo Sensor State
  + Date/Time
  + Fuel Level
  + Temperature
* IT
  + System Interrupts per Second
  + Processor Performance
  + User Time
  + Clock Interrupts per second

**Discussion [Brett Wolff]**

This lab builds upon what was done in the previous status update. This is continuing to add more data sources to the workflow to primarily focus on improving the data model and overall accuracy. Having an increased amount of data means that when the algorithm is making a prediction, it should be more accurate. Adding data like the processor performance could clue the algorithm into something that could be stealing resources from the systems, which would help detect any intrusions or persistence. Adding alongside this is learning how to use the IT data and insert it into the system. As mentioned previously, the way the data was saved made it difficult to adjust and read certain parts. However, learning about how to use this data means that the teams will know how to add more IT information in the near future.

**Conclusion [Brett Wolff]**

The team expanded the amount of data in the workflow by adding a couple more sources of IT data and OPTO data. This was done with the ultimate goal of having a more accurate algorithm and to be able to detect abnormalities with higher accuracy. This also leaves the workflow in a spot that could allow for more data sources once the data added here is fully integrated.