ARL-TR-9250 ● July 2021

Access and Usage of the Statistical Analysis System in Secured Unclassified Network (SUNet)

by Sean Harding and Kevin Chan

Approved for public release: distribution unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer’s or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9250 ● July 2021

Access and Usage of the Statistical Analysis System in Secured Unclassified Network (SUNet)

Sean Harding
ICF International

Kevin Chan
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
July 2021

2. REPORT TYPE
Technical Report

3. DATES COVERED (From - To)
September 2020–January 2021

4. TITLE AND SUBTITLE
Access and Usage of the Statistical Analysis System in Secured Unclassified Network (SUNet)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sean Harding and Kevin Chan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD-RLC-NT
2800 Powder Mill Rd
Adelphi, MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9250

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report details the creation and usage of a Secured Unclassified Network (SUNet) account in order to use the tools offered by that system, including the Statistical Analysis System tool suite.

15. SUBJECT TERMS
Secured Unclassified Network, SUNet, Statistical Analysis System, SAS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Sean Harding
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	18	**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified			(301) 394-4059

Contents

List of Figures

Summary

This report will establish a basic understanding of how to access and use the basic functions of the Statistical Analysis System (SAS) via Secured Unclassified Network (SUNet) as well as how to bring your own algorithms and have them integrated into the system. We recommend contacting Army Artificial Intelligence Institute (A2I2) for access to this platform.

1. Introduction

The Statistical Analysis System (SAS) is a software suite designed to assist in data discovery, exploration, and analysis. Access to SAS can be gained through signing into Kasm and creating a new image. You then can access SAS either via a graphical user interface (GUI) or through the command line. Using these tools you can manipulate data, generate models, and analyze those models.

2. Methods, Assumptions, and Procedures

Gaining access to SAS is a multistep process that begins by being granted the necessary accesses and privileges.

2.1 Kasm

To access the Kasm system, navigate to the https://kasm-cac.761link.net. Once accessed, log in with your Common Access Card (CAC) and credentials as shown in Fig. 1. You will now find yourself in the Kasm desktop, which will give you access to SAS tools.

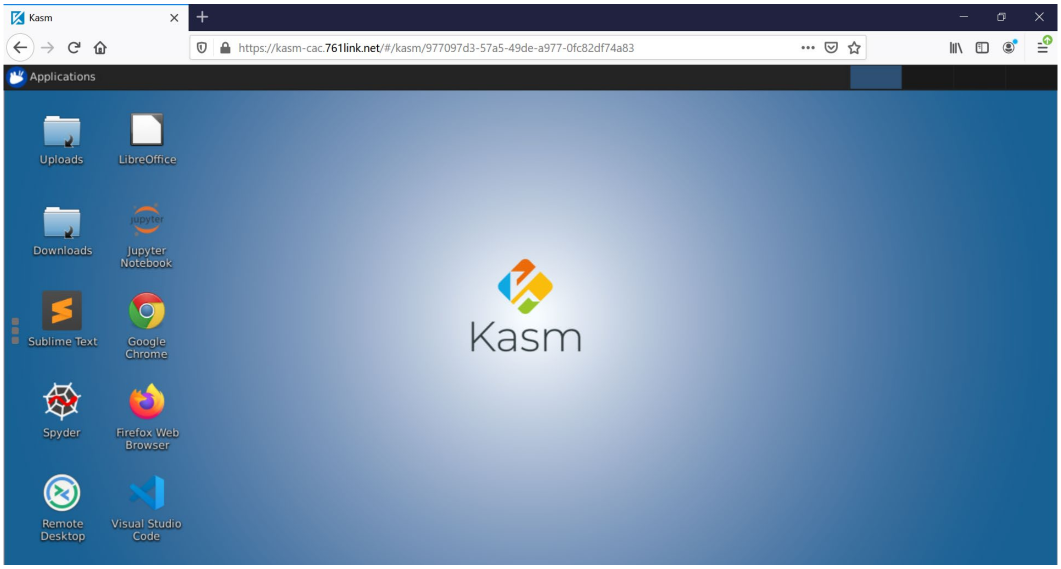


Fig. 1 Kasm desktop

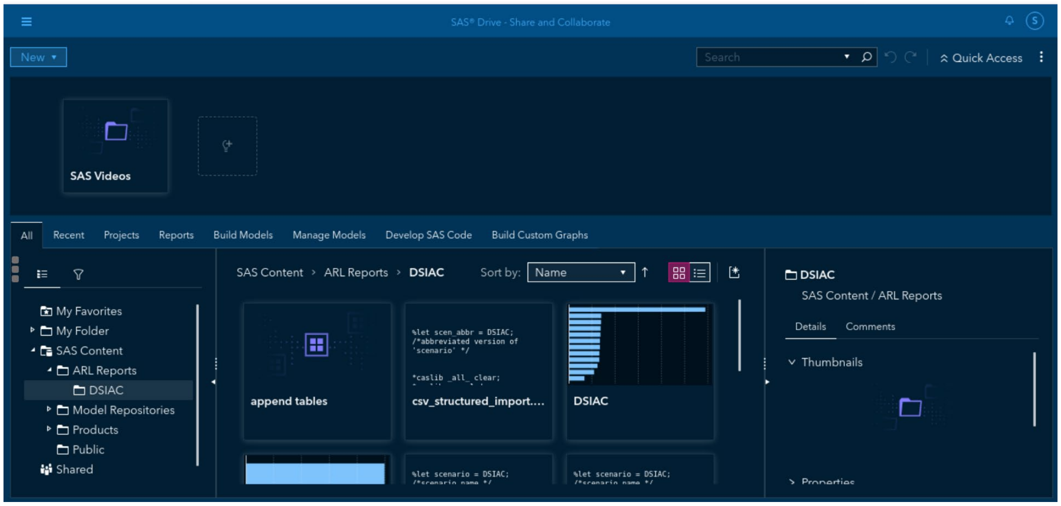
2.2 SAS Logon

Now that you have access to the Kasm desktop, you can access SAS. To do this, open a browser (Chrome is suggested), proceed to https://a2i2-sas-vm-ms1.omaha.internal/SASLogon, and use your provided SAS credentials to gain access.

3. Results and Discussion

3.1 SAS Drive

This functionality acts as the file manager and allows for accessing files and analyzing the contents of those files. You can select data here to use in other tools, such as the SAS Visual Analytics tool, as shown in Fig. 2.



```
%let scen_abbr = DSIAC;
/*abbreviated version of
'scenario' */

*caslib _all_ clear;
```

```
%let scenario = DSIAC;
/*scenario name */
```

```
%let scenario = DSIAC;
/*scenario name */
```

Fig. 2 SAS drive

3.2 SAS Studio

Figure 3 shows the integrated development environment within SAS.

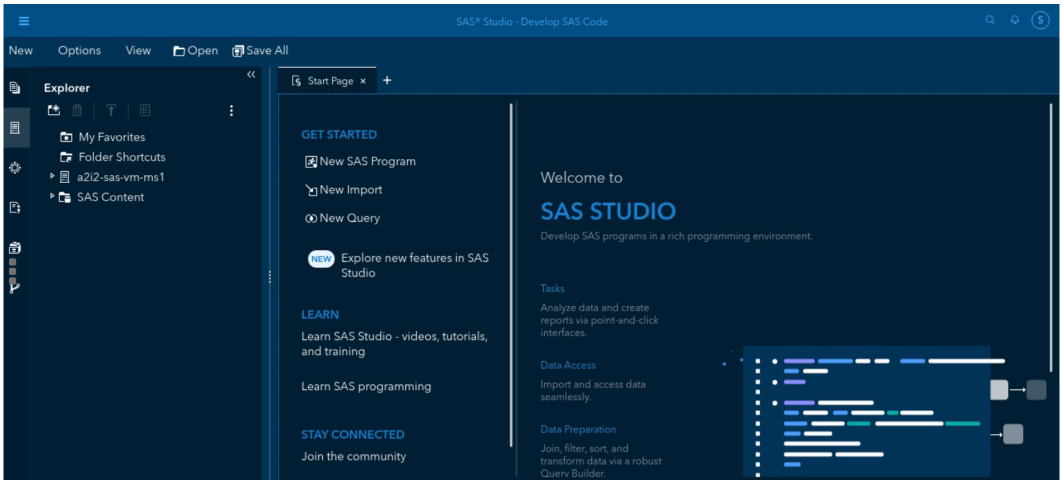

SAS® Studio - Develop SAS Code
New
Options
View
Open
Save All
Explorer
My Favorites
Folder Shortcuts
a2i2-sas-vm-ms1
SAS Content
Start Page ×
+
GET STARTED
New SAS Program
New Import
New Query
NEW
Explore new features in SAS Studio
LEARN
Learn SAS Studio - videos, tutorials, and training
Learn SAS programming
STAY CONNECTED
Join the community
Welcome to
SAS STUDIO
Develop SAS programs in a rich programming environment.
Tasks
Analyze data and create reports via point-and-click interfaces.
Data Access
Import and access data seamlessly.
Data Preparation
Join, filter, sort, and transform data via a robust Query Builder.

Fig. 3 SAS studio

3.3 SAS Visual Analytics

Visual Analytics, shown in Fig. 4, allows you to choose a data source and analyze the individual features and entries within those sources.

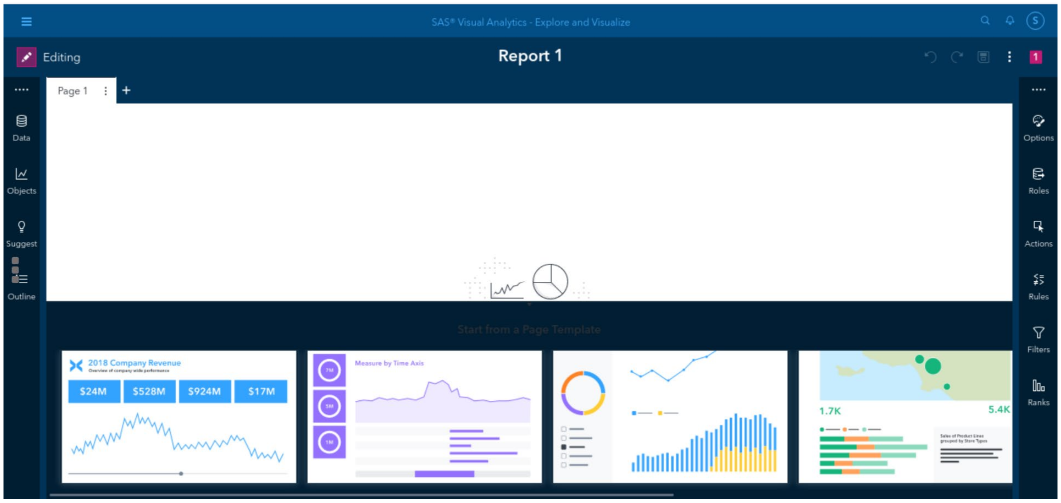
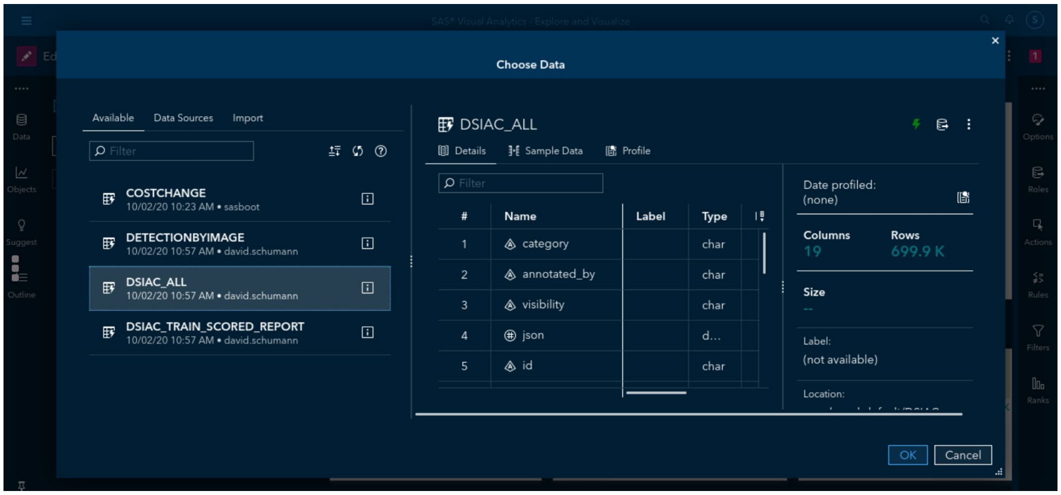


Fig. 4 Visual Analytics main page

You can also create reports and visualizations based on the chosen data, as shown in Fig. 5.



COSTCHANGE
10/02/20 10:23 AM • sasboot

DETECTIONBYIMAGE
10/02/20 10:57 AM • david.schumann

DSIAC_ALL
10/02/20 10:57 AM • david.schumann

DSIAC_TRAIN_SCORED_REPORT
10/02/20 10:57 AM • david.schumann

DSIAC_ALL

#	Name	Label	Type
1	category		char
2	annotated_by		char
3	visibility		char
4	json		d...
5	id		char

Date profiled:
(none)

Columns 19
Rows 699.9 K

Size
--

Label:
(not available)

Location:

Fig. 5 Visual Analytics loading data

3.4 Command Line Interface

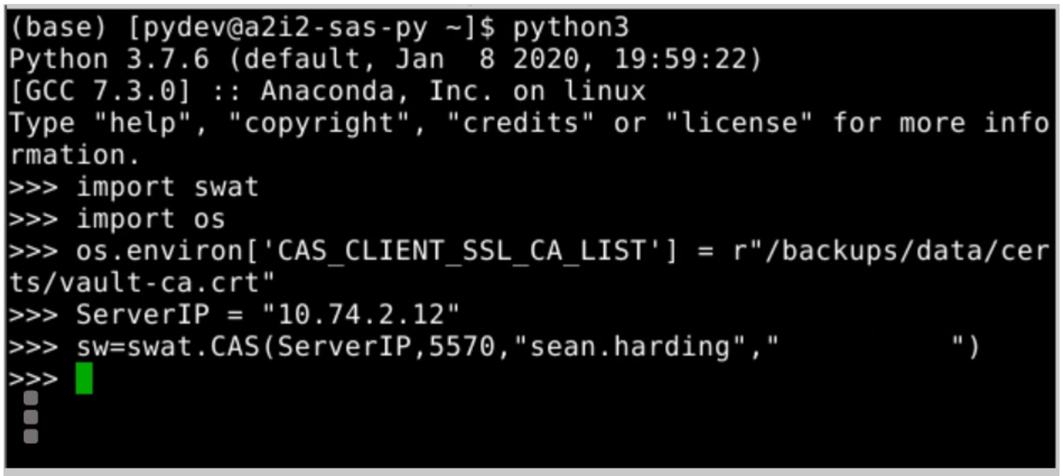
You can also access SAS through the command line interface (CLI). To do this you must first log on to Kasm as normal. Next, open a terminal and ssh into username@10.74.2.5 using your provided credentials. Once here, you will have access to the SAS backend including libraries such as Scripting Wrapper for Analytics Transfer (SWAT)[1] among many others.

An example of gaining access to CLI SAS and using the SWAT library:

```
from swat import *
import pandas as pd
import os
#Select the SSL certificate for connecting to SAS
os.environ['CAS_CLIENT_SSL_CA_LIST'] =
'/opt/sas/viya/certs/vault-ca.crt'
#Connect to CAS Server
ServerIP="10.74.2.12"
sw=CAS(ServerIP,5570,"username-here","password-
here")
#Loop Through a list of actions and load them all
actions=['image','cardinality','fedsql','ds2','sccasl'
,'transpose']
[sw.builtins.loadactionset(i) for i in actions]
```

3.4.1 Available Libraries

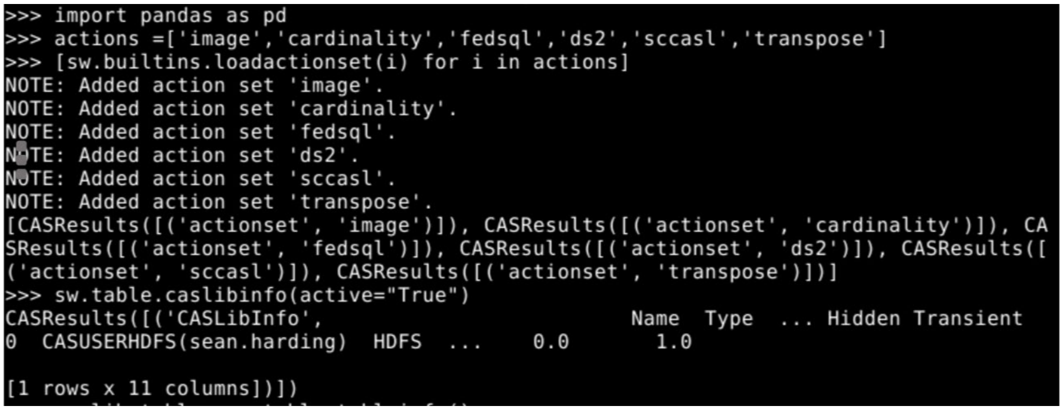
In the CLI, shown in Fig. 6, you can use SWAT in order to access SAS tools. To do this you simply need to use Python 3 and import the SWAT library. SWAT grants access to additional tools such as the Cloud Analytic Services (CAS).



```
(base) [pydev@a2i2-sas-py ~]$ python3
Python 3.7.6 (default, Jan  8 2020, 19:59:22)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more info
rmation.
>>> import swat
>>> import os
>>> os.environ['CAS_CLIENT_SSL_CA_LIST'] = r"/backups/data/cer
ts/vault-ca.crt"
>>> ServerIP = "10.74.2.12"
>>> sw=swat.CAS(ServerIP,5570,"sean.harding","          ")
>>> 
```

Fig. 6 Accessing SAS from CLI

Figure 7 displays the correct way to access the CAS library. This grants you command line access to use SAS tools and allows you to access information from SAS Drive. Ensure that you use your own username and password in order to access CAS.

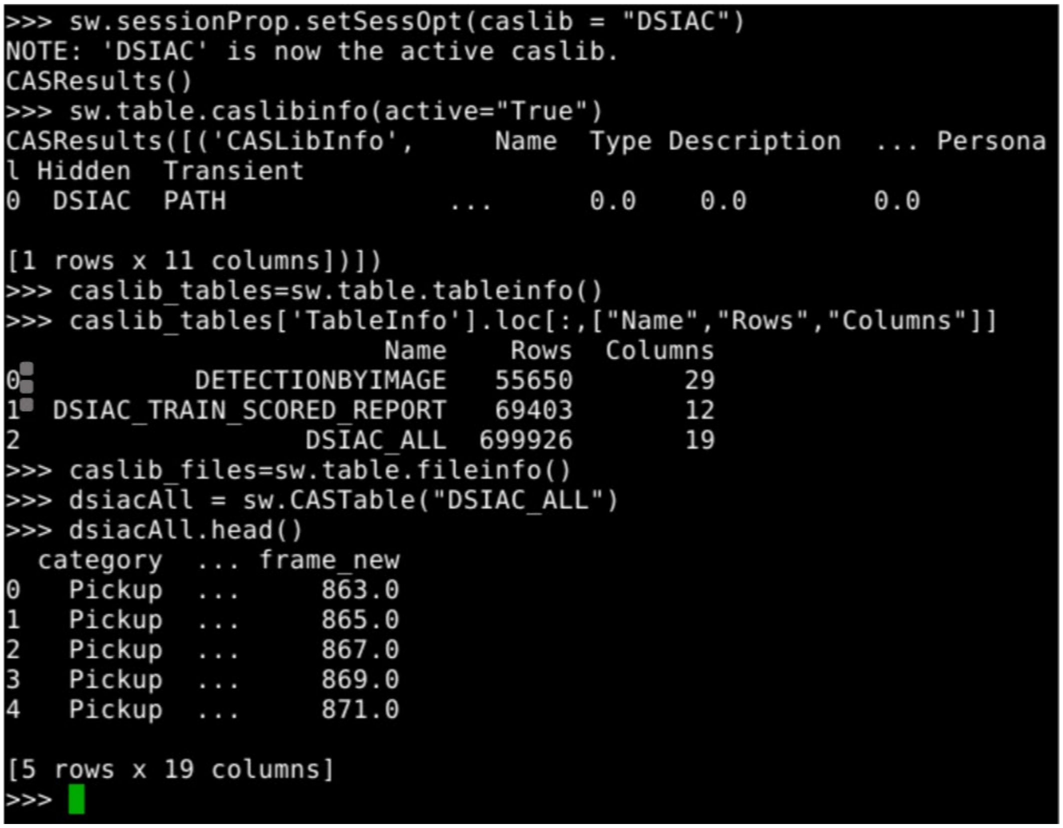


```
>>> import pandas as pd
>>> actions =['image','cardinality','fedsql','ds2','sccasl','transpose']
>>> [sw.builtins.loadactionset(i) for i in actions]
NOTE: Added action set 'image'.
NOTE: Added action set 'cardinality'.
NOTE: Added action set 'fedsql'.
NOTE: Added action set 'ds2'.
NOTE: Added action set 'sccasl'.
NOTE: Added action set 'transpose'.
[CASResults([('actionset', 'image')]), CASResults([('actionset', 'cardinality')]), CA
SResults([('actionset', 'fedsql')]), CASResults([('actionset', 'ds2')]), CASResults([
('actionset', 'sccasl')]), CASResults([('actionset', 'transpose')])]
>>> sw.table.caslibinfo(active="True")
CASResults([('CASLibInfo',                         Name  Type  ... Hidden Transient
0  CASUSERHDFS(sean.harding)  HDFS  ...    0.0       1.0

[1 rows x 11 columns])])
```

Fig. 7 Accessing data from SAS drive

In the example in Fig. 8, a set of actions have been added that will allow us to manipulate data we load in.



```
>>> sw.sessionProp.setSessOpt(caslib = "DSIAC")
NOTE: 'DSIAC' is now the active caslib.
CASResults()
>>> sw.table.caslibinfo(active="True")
CASResults([('CASLibInfo',     Name  Type Description  ... Persona
l Hidden  Transient
0  DSIAC  PATH              ...      0.0    0.0        0.0

[1 rows x 11 columns])])
>>> caslib_tables=sw.table.tableinfo()
>>> caslib_tables['TableInfo'].loc[:,["Name","Rows","Columns"]]
                        Name    Rows  Columns
0           DETECTIONBYIMAGE   55650       29
1  DSIAC_TRAIN_SCORED_REPORT   69403       12
2                  DSIAC_ALL  699926       19
>>> caslib_files=sw.table.fileinfo()
>>> dsiacAll = sw.CASTable("DSIAC_ALL")
>>> dsiacAll.head()
  category  ... frame_new
0   Pickup  ...     863.0
1   Pickup  ...     865.0
2   Pickup  ...     867.0
3   Pickup  ...     869.0
4   Pickup  ...     871.0

[5 rows x 19 columns]
>>> 
```

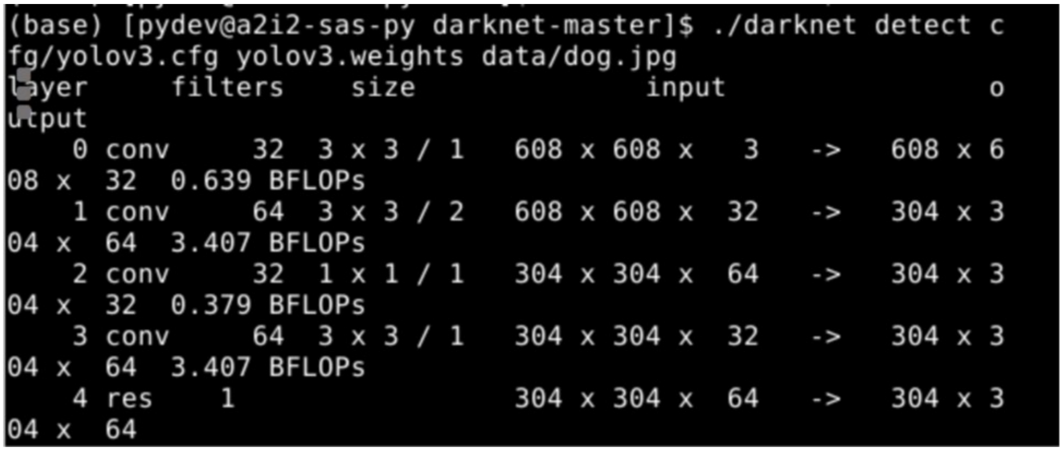
Fig. 8   Loading data from SAS drive

In Fig. 8, we load data in and access it. Following this methodology you can work with any data in SAS or add data in.

3.5 Integrating Algorithms

Algorithms you wish to integrate into the SAS development environment will need to be ported over from your existing systems. To do this, give your Secured Unclassified Network (SUNet) contact a copy of the materials you want ported over. This process can take a few days so ensure that you give yourself plenty of time before needing this algorithm on the system. Once your algorithm has been introduced into the system you can integrate it into the SAS suite through the use of the SWAT library in the CLI we looked at previously. This will allow you to take advantage of the tools that the GUI offers.

As you can see in Fig. 9, a custom algorithm is running on the system, in this case Darknet,[2] an object detection algorithm. After running our algorithm and getting results we can analyze the data created, things such as models created and weights trained or objects detected in the SAS system, by accessing those tools as we have done previously through the CLI.



```
(base) [pydev@a2i2-sas-py darknet-master]$ ./darknet detect c
fg/yolov3.cfg yolov3.weights data/dog.jpg
layer     filters    size              input                o
utput
    0 conv     32  3 x 3 / 1   608 x 608 x   3   ->   608 x 6
08 x  32  0.639 BFLOPs
    1 conv     64  3 x 3 / 2   608 x 608 x  32   ->   304 x 3
04 x  64  3.407 BFLOPs
    2 conv     32  1 x 1 / 1   304 x 304 x  64   ->   304 x 3
04 x  32  0.379 BFLOPs
    3 conv     64  3 x 3 / 1   304 x 304 x  32   ->   304 x 3
04 x  64  3.407 BFLOPs
    4 res    1                 304 x 304 x  64   ->   304 x 3
04 x  64
```

Fig. 9 Running a custom algorithm

4. Discussion and Conclusion

SAS offers a multitude of ways to interact and evaluate your data and SAS through SUNet promises the extendibility of being able to bring in your own algorithms and analyze the data generated by those algorithms. However, some potential sticking points do exist. The login process to accessing SAS, especially in the CLI, is cumbersome. This could be alleviated by creating a single sign-on between Kasm and the SAS server. While the CLI is overall a useful tool for developers, SAS's primary target user is an analyst using the SAS GUI, and as such the CLI can sometimes feel like an afterthought. Lastly, the process of moving data and code off or on the SUNet systems requires the assistance of an administrator and, depending on the size of the data, quite a bit of time as well. Some solutions may involve automating the file transfer process to remove the lag time that communicating through email inevitably brings.

5. References

1. SAS Software. Scripting wrapper for analytics transfer [accessed 2021 Jan 27]. https://github.com/sassoftware/python-swat.

2. Redmon J. Yolo: real time object detection [accessed 2021 Jan 27]. https://pjreddie.com/darknet/yolo/.

List of Symbols, Abbreviations, and Acronyms

A2I2	Army Artificial Intelligence Institute
ARL	Army Research Laboratory
CAC	Common Access Card
CAS	Cloud Analytic Services
CLI	command line interface
DEVCOM	US Army Combat Capabilities Development Command
GUI	graphical user interface
SAS	Statistical Analysis System
SUNet	Secured Unclassified Network
SWAT	Scripting Wrapper for Analytics Transfer

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLD DCI
TECH LIB

2 DEVCOM ARL
(PDF) FCDD RLC NT
S HARDING
K CHAN