black hat BRIEFINGS

AUGUST 6-7, 2025
MANDALAY BAY / LAS VEGAS

Anomaly Detection Betrayed Us, so We Gave It a New Job: Enhancing Command Line Classification with Benign Anomalous Data

Ben Gelman, Sean Bergeron

Introduction

About Me - Ben

- Data Scientist at Sophos for 4 years

- 5 years in government-funded R&D

- 2 years of post-grad research at academic institutions

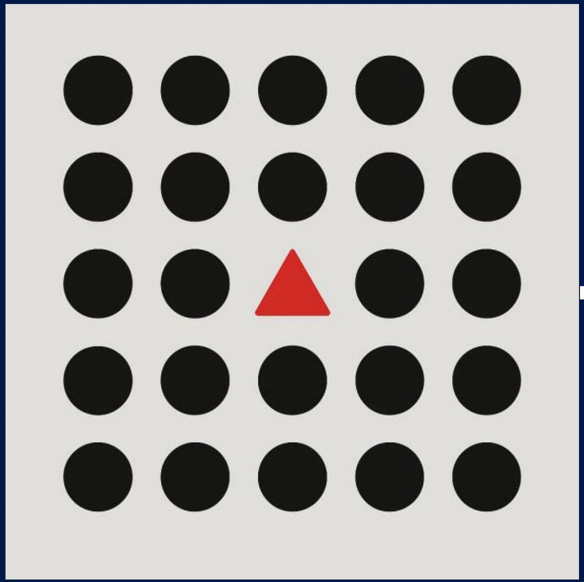
About Me - Sean

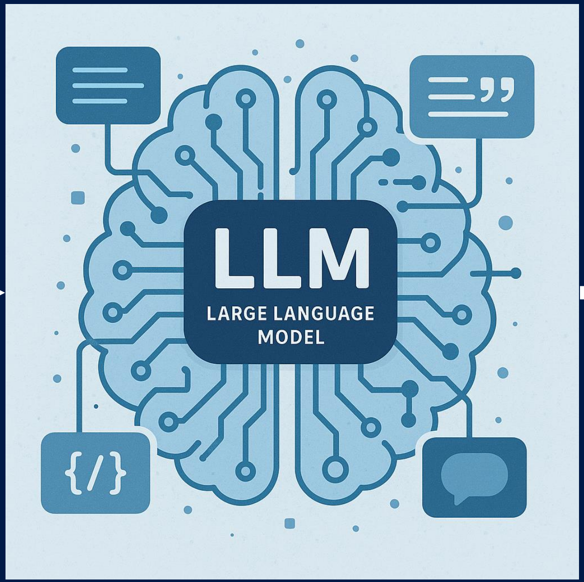
- Data Scientist at Sophos for 3 years

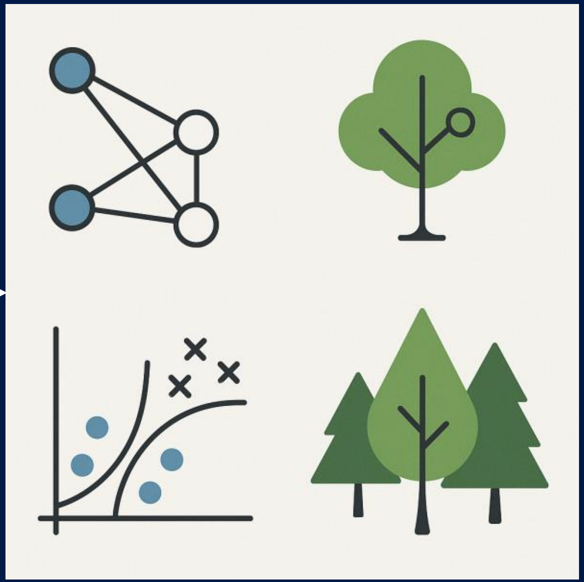
- Deep personality estimation post-grad research

- Mechanical engineer

What Are We Talking About?

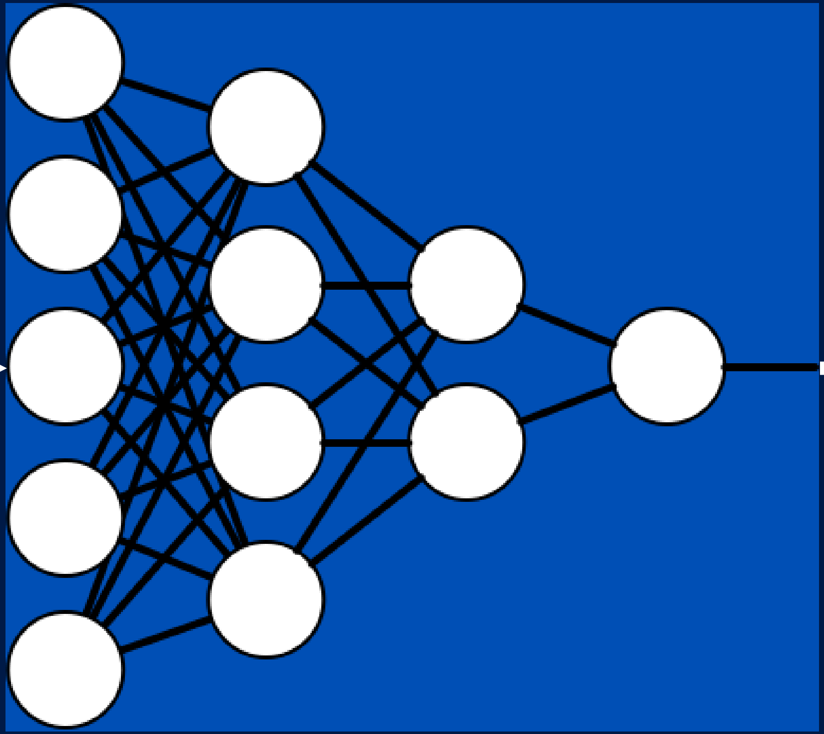


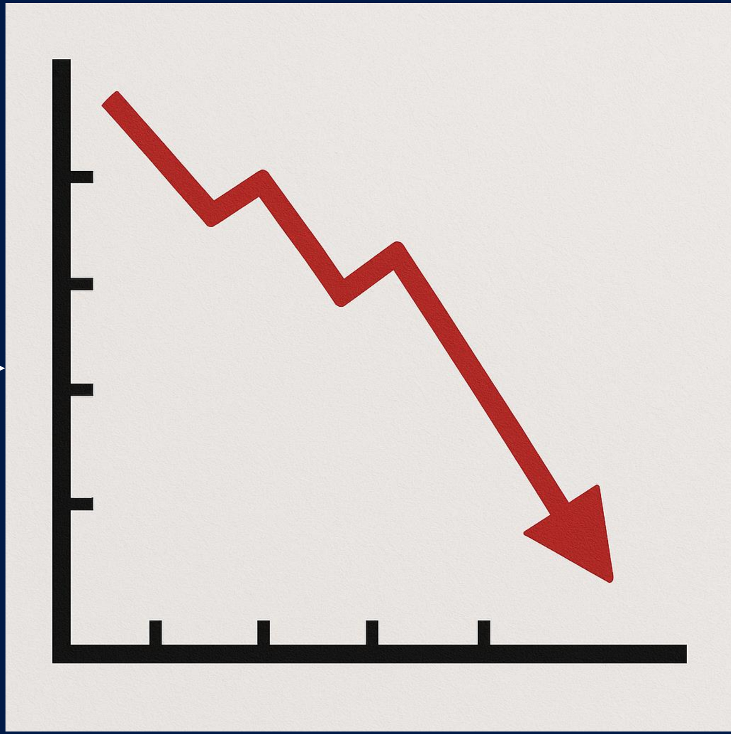




How did this happen?

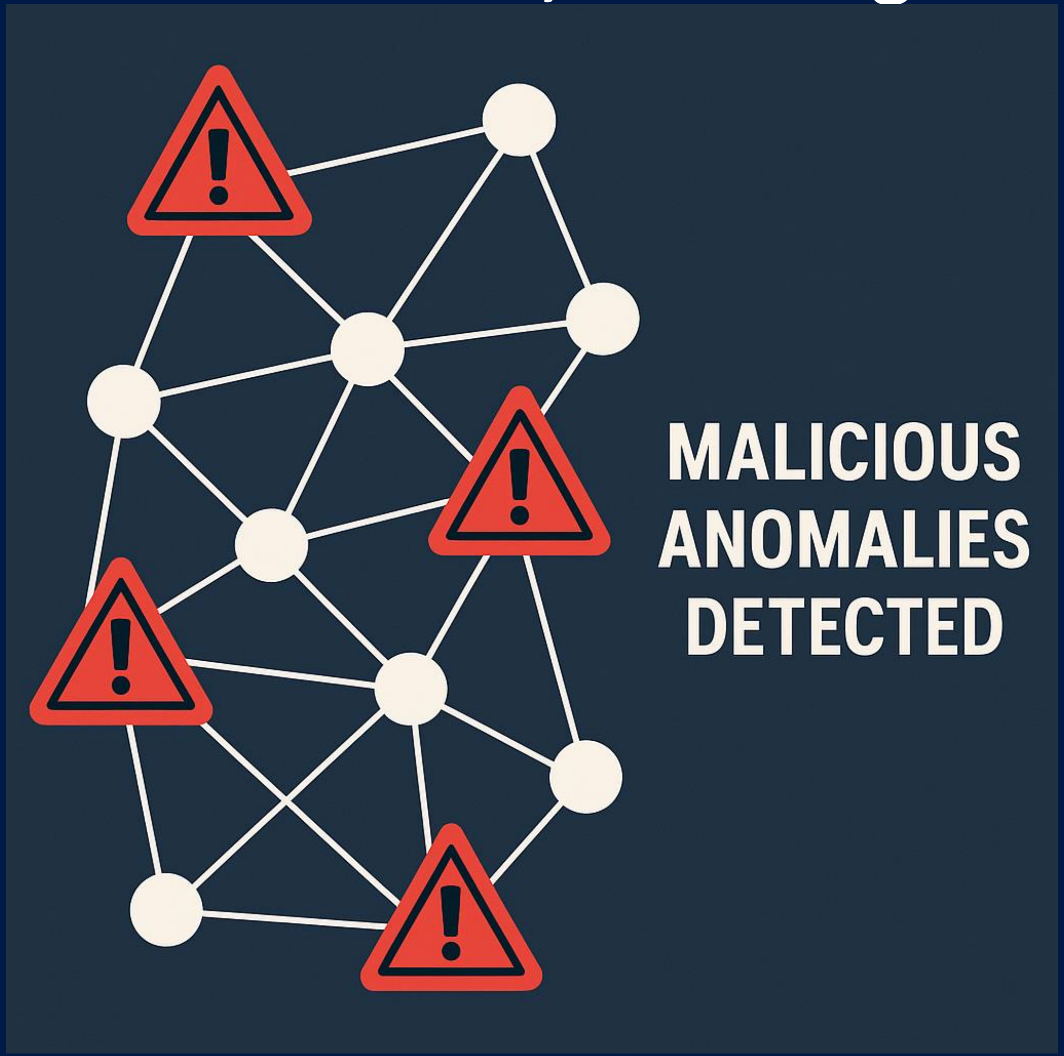
Command lines



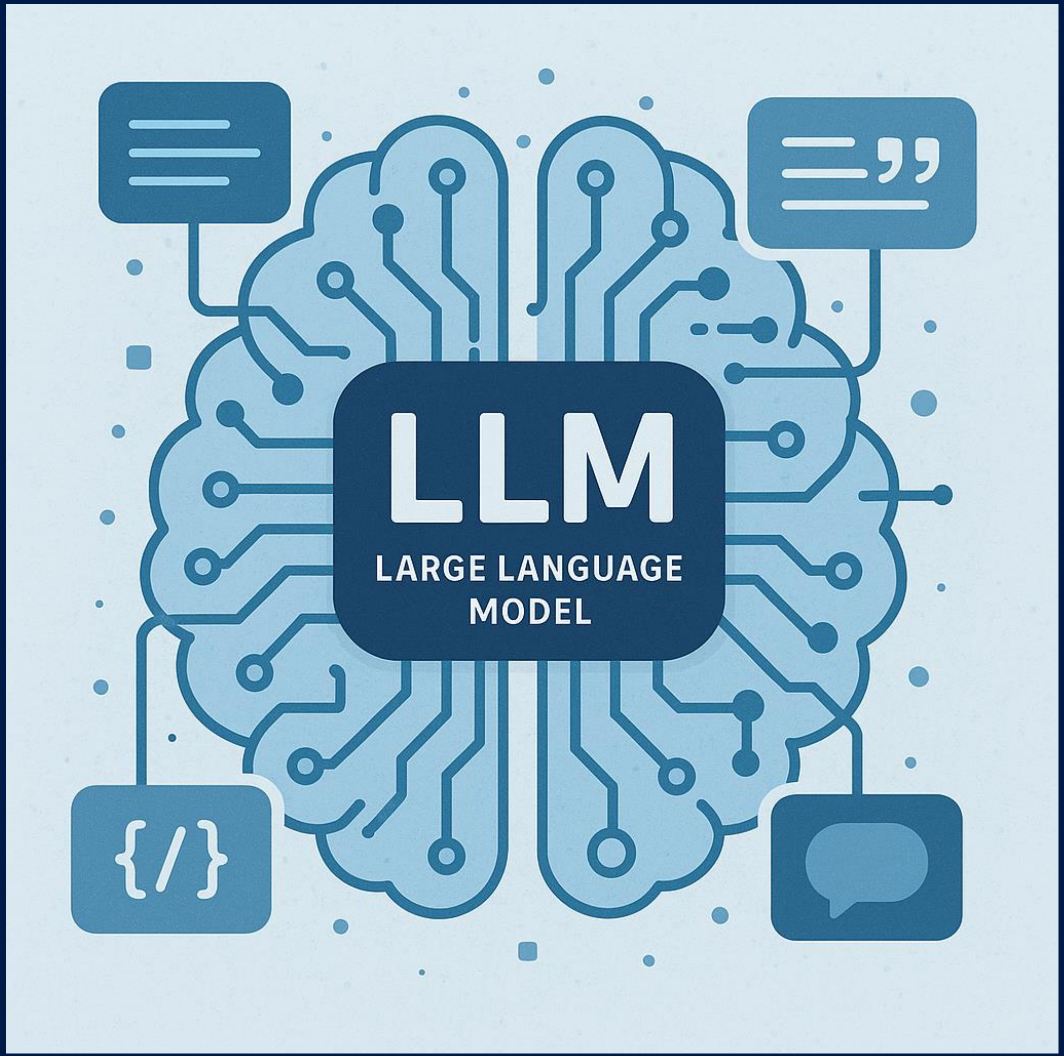


Unsustainable Manual Effort

The Perfect, Fully-Automated, Self-Updating System for Command Line Prediction, Featuring LLMs™



+



Not Really: Anomaly Detection Betrayed Us

36%

Malicious Precision

100%

Benign Precision

Motivation

Unsupervised: The State of Anomaly Detection

Pros

- No labels required
- High scalability
- Low Cost

Cons

- High false positive rates – extreme alert fatigue
- Reliance on human expertise

The State of Anomaly Detection

FPR <2, <2, <3%

[1, 2, 3]

Feasible?

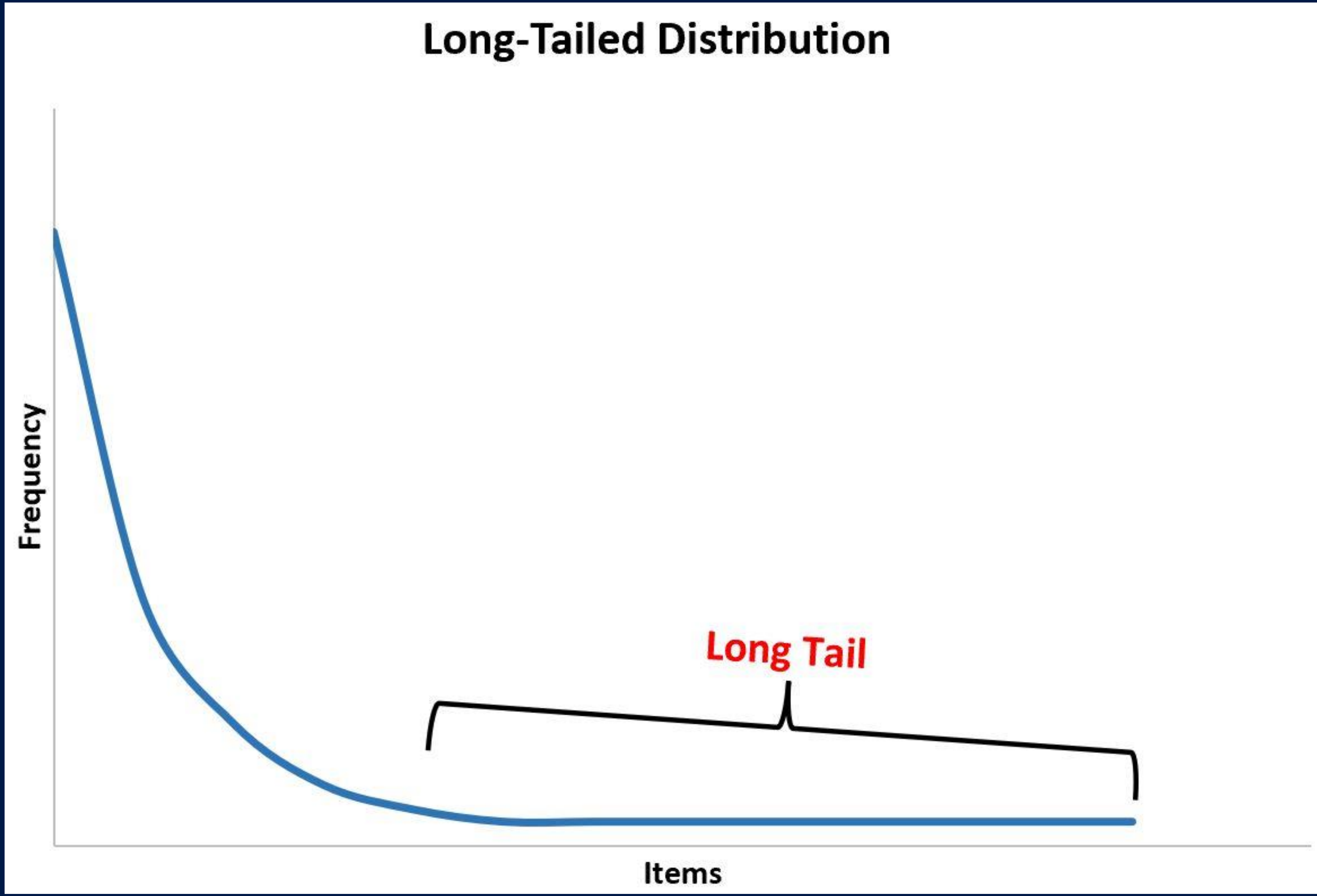
The State of Anomaly Detection

The State of Labeled Data: Malicious Data

- Sandbox
- VirusTotal
- Customer Case Investigations
- Expert labeling

The State of Labeled Data: Benign Data [4, 5]

The Longtail



Are We Stuck?

Anomaly Detection

High FP Rates

Scalable

Supervised Learning

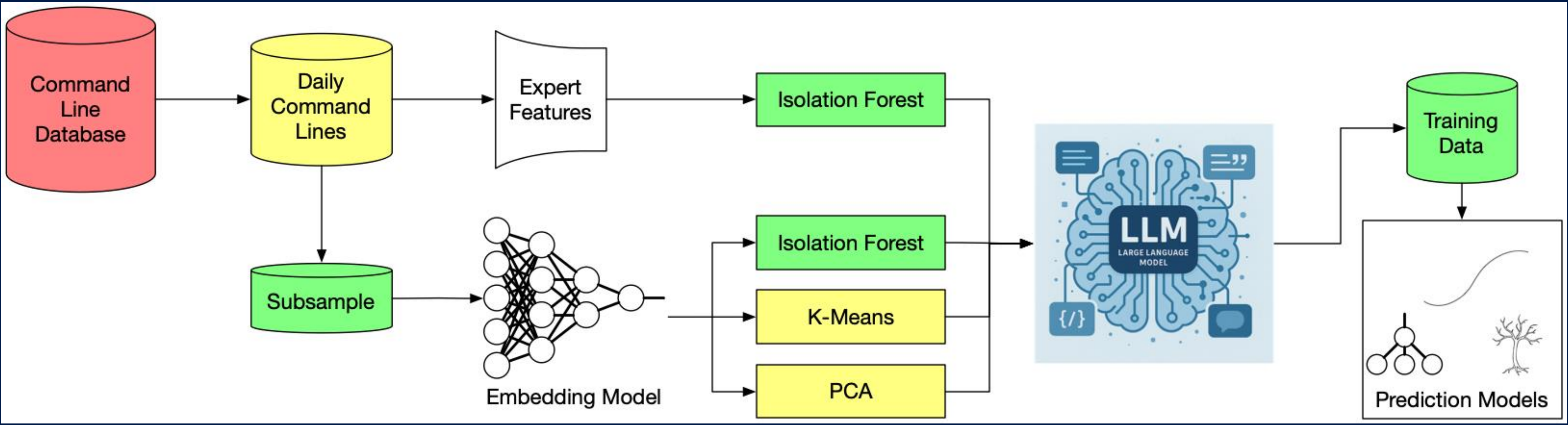
Scalability Issues

Low FP Rates

Are We Stuck?

Redefining The Role of Anomaly Detection

The Whole System



Command Line Datasets

1.) Regex-based dataset

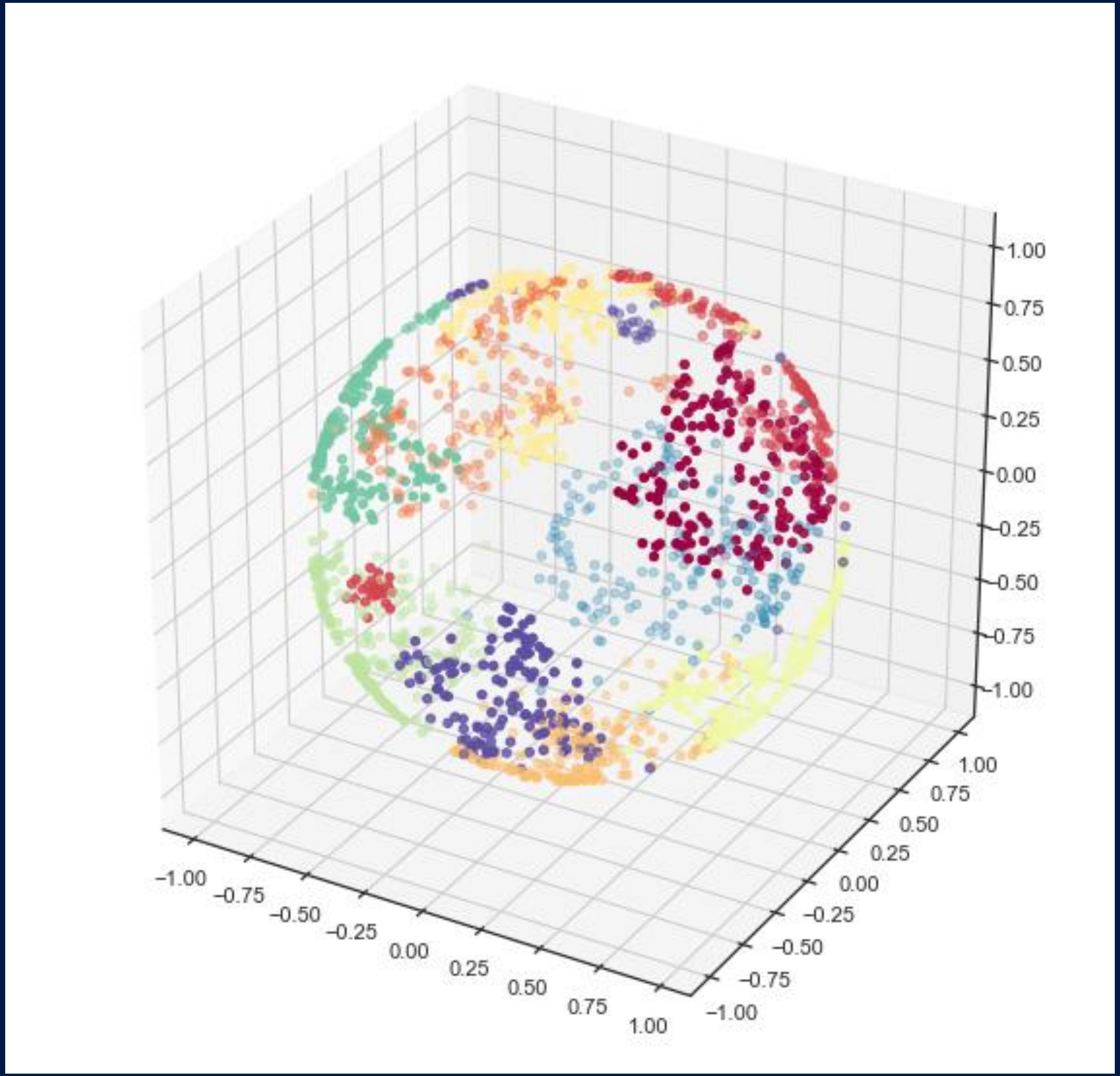
2.) Aggregated dataset

- Regex
- Sandbox
- Case Investigations
- Customer telemetry

Expert Features

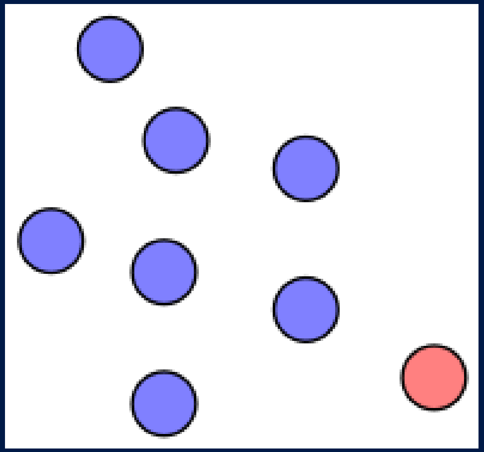
$$H = -\sum p(x) \log p(x)$$

Embeddings Model

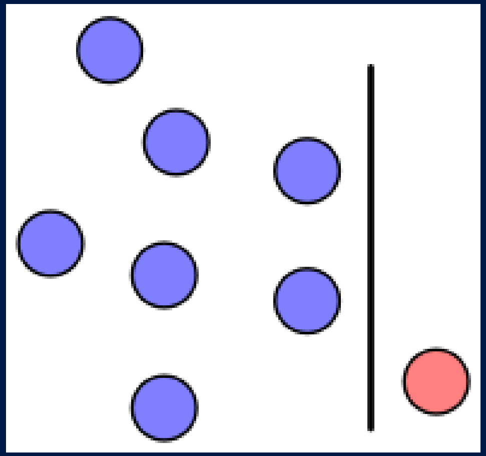


Isolation Forest

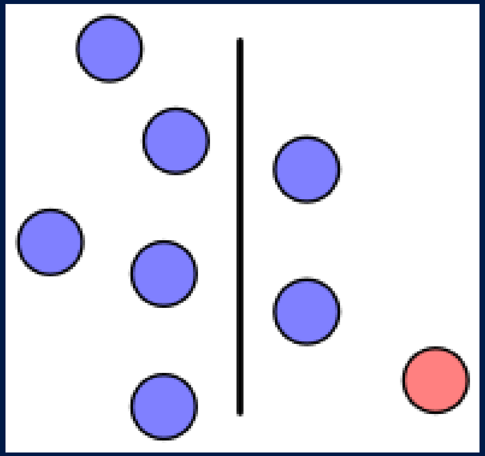
1



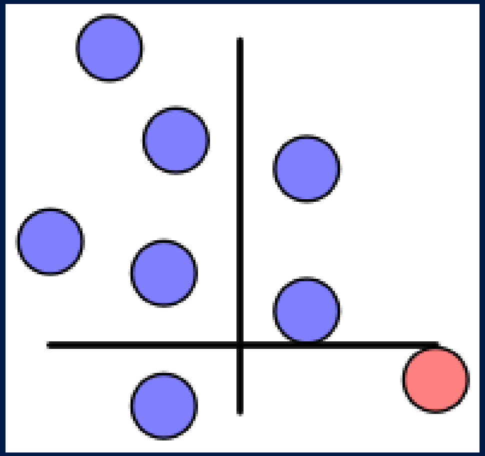
2



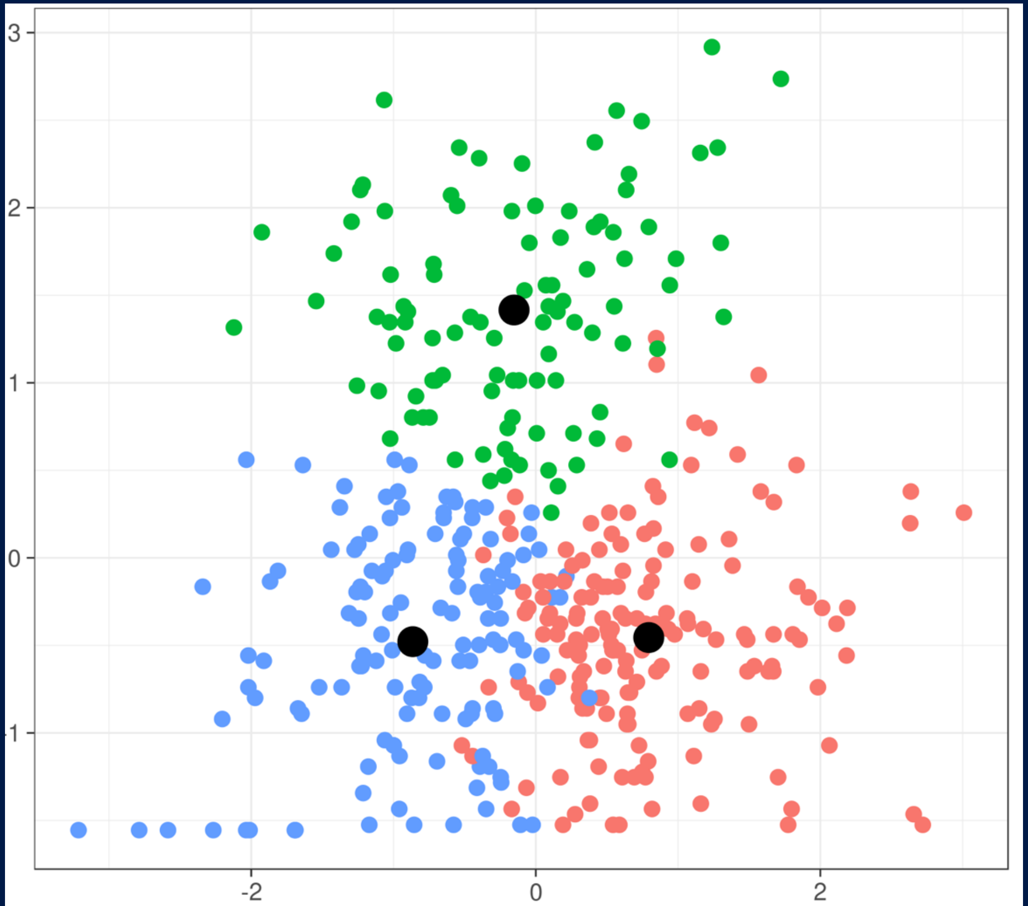
3

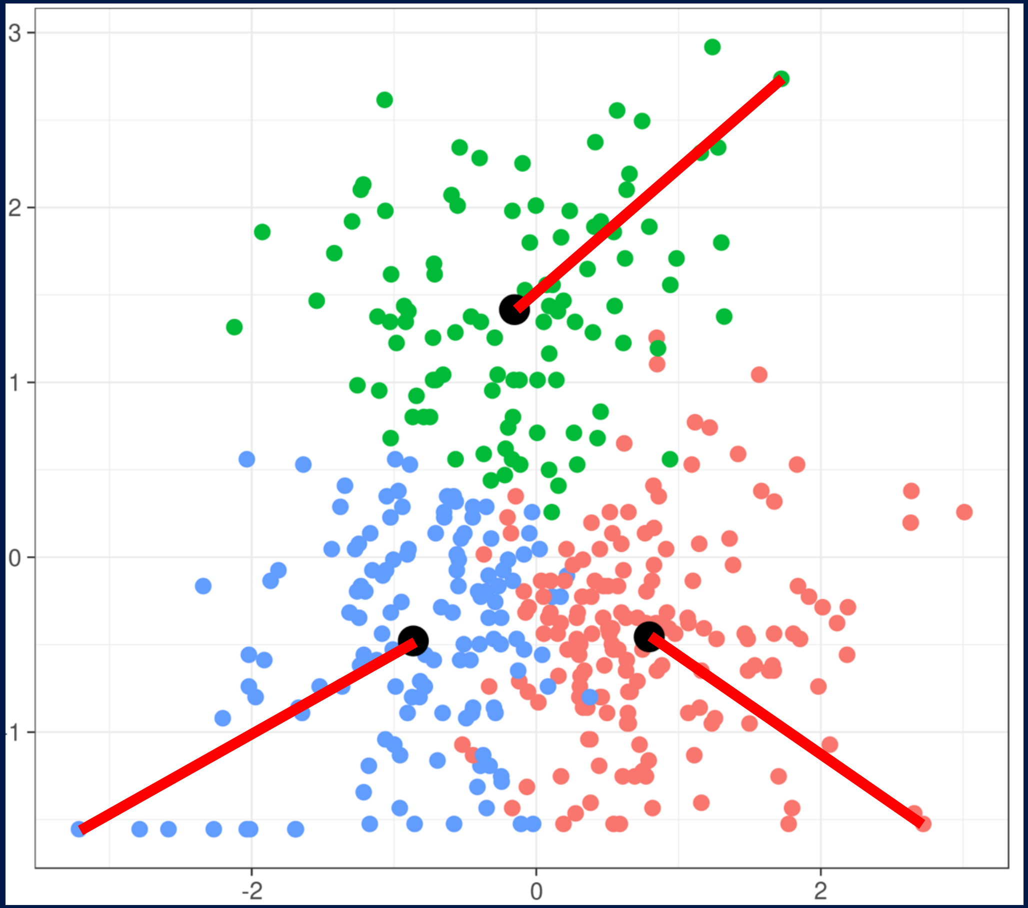


4

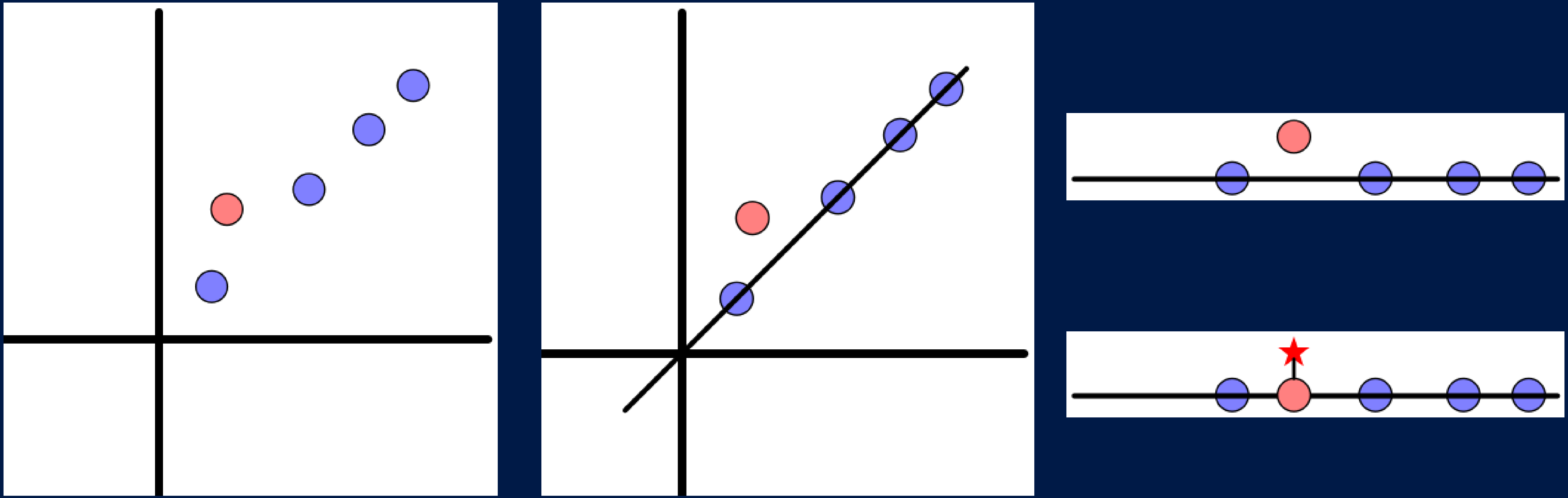


K-means Anomaly





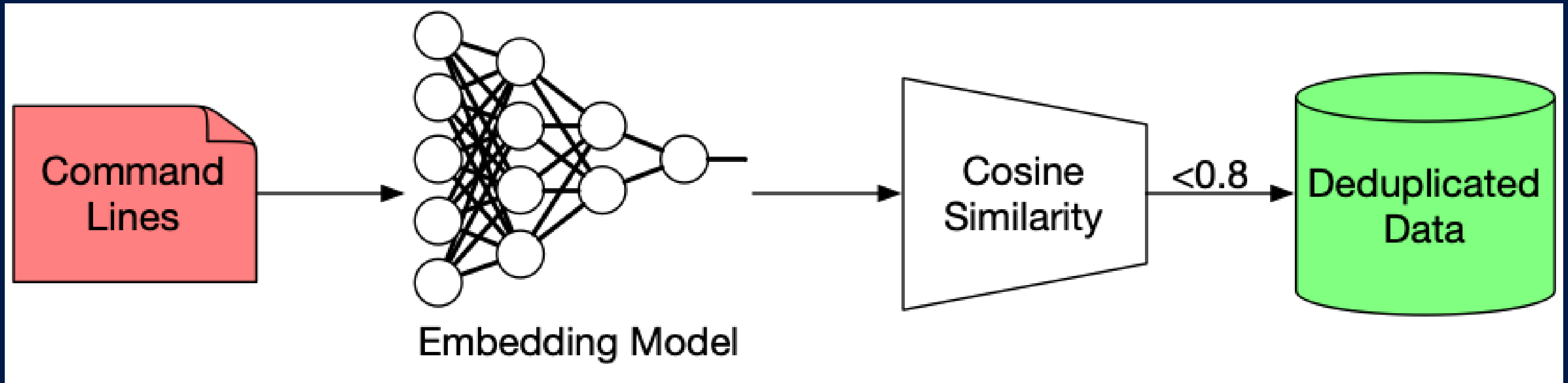
Principal Components Analysis (PCA) Anomaly



Deduplication

- Two nearly duplicate command lines:
  - ls ***-l*** /home/user
  - ls ***-la*** /home/user

Exact Deduplication			Near Deduplication		
Command 1	Command 2	Dedupe?	Command 1	Command 2	Dedupe?
ls -l /home/user	ls -l /home/user	✅	ls -l /home/user	ls -l /home/user	✅
ls -l /home/user	ls -la /home/user	❌	ls -l /home/user	ls -la /home/user	✅



LLM Labeling

LLM Labeling + Demo

- "E:\Applications\AODB\Java\jdk1.8.0_412\bin\aodb-java" -Dprogram.name=standalone.bat -server -XX:MaxMetaspaceSize=1024M -XX:MetaspaceSize=1024M -XX:+UseParallelGC -XX:+UseParallelOldGC -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.net.preferIPv4Stack=true -Dorg.tanukisoftware.wrapper.WrapperManager.mbean=false -Djboss.server.default.config=standalone.xml -Dlogging.configuration=file:E:\Applications\AODB\wildfly-26.1.6.Final/standalone/configuration/logging.properties -Dorg.jboss.boot.log.file=E:\Applications\AODB\wildfly-26.1.6.Final/standalone/log/boot.log -Djava.util.logging.manager=org.jboss.logmanager.LogManager -Dorg.jboss.logging.Logger.pluginClass=org.jboss.logging.logmanager.LoggerPluginImpl -Djboss.remoting.pooled-buffers=false -Dfile.encoding=Cp1252 -Duser.language=en -Xms2048m -Xmx16384m -Djava.library.path="E:\Applications\AODB\wildfly-26.1.6.Final\lib" -classpath "E:\Applications\AODB\wildfly-26.1.6.Final\lib\wrapper.jar;E:\Applications\AODB\wildfly-26.1.6.Final\jboss-modules.jar" -Dwrapper.key="v19OywX5EMaygmtSZdG9t35Naj6wvoH9" -Dwrapper.port=32000 -Dwrapper.jvm.port.min=31000 -Dwrapper.jvm.port.max=31999 -Dwrapper.debug="TRUE" -Dwrapper.pid=2008 -Dwrapper.version="3.5.25-pro" -Dwrapper.native_library="wrapper" -Dwrapper.arch="x86" -Dwrapper.service="TRUE" -Dwrapper.cpu.timeout="10" -Dwrapper.jvmid=2 -Dwrapper.lang.domain=wrapper -Dwrapper.lang.folder=../lang org.tanukisoftware.wrapper.WrapperJarApp jboss-modules.jar -mp E:\Applications\AODB\wildfly-26.1.6.Final/modules org.jboss.as.standalone -Djboss.home.dir=E:\Applications\AODB\wildfly-26.1.6.Final --server-config=standalone-full-sqlsrv.xml -P=E:\Applications\AODB\wildfly-26.1.6.Final/standalone/configuration/aodb-sqlsrv.properties -b 192.168.7.61

User

We need to determine whether a command line is benign or malicious.
We have to be very confident in our answer, so let's think about this deeply.
Let's start by explaining what the command line does.
Then explain how, if possible, the command could be used in a benign way.
Then explain how, if possible, the command line could be used maliciously.
Finally, make a verdict on whether the command line is benign or malicious.
Output your response as either VERDICT_BENIGN or VERDICT_MALICIOUS.
Once again, only convict a command as malicious if it is very likely.
Use the term VERDICT_BENIGN or VERDICT_MALICIOUS only ONCE in your response.
Command line: "E:\Applications\AODB\Java\jdk1.8.0_412\bin\aodb-java" -Dprogram.name=standalone.bat -server -XX:MaxMetaspaceSize=1024M -XX:MetaspaceSize=1024M -XX:+UseParallelGC -XX:+UseParallelOldGC -Dsun.rmi.dgc.client.gcInterval=3600000 -

Collapse

Chat with your prompt...

Auto-clear

Results

Evaluation

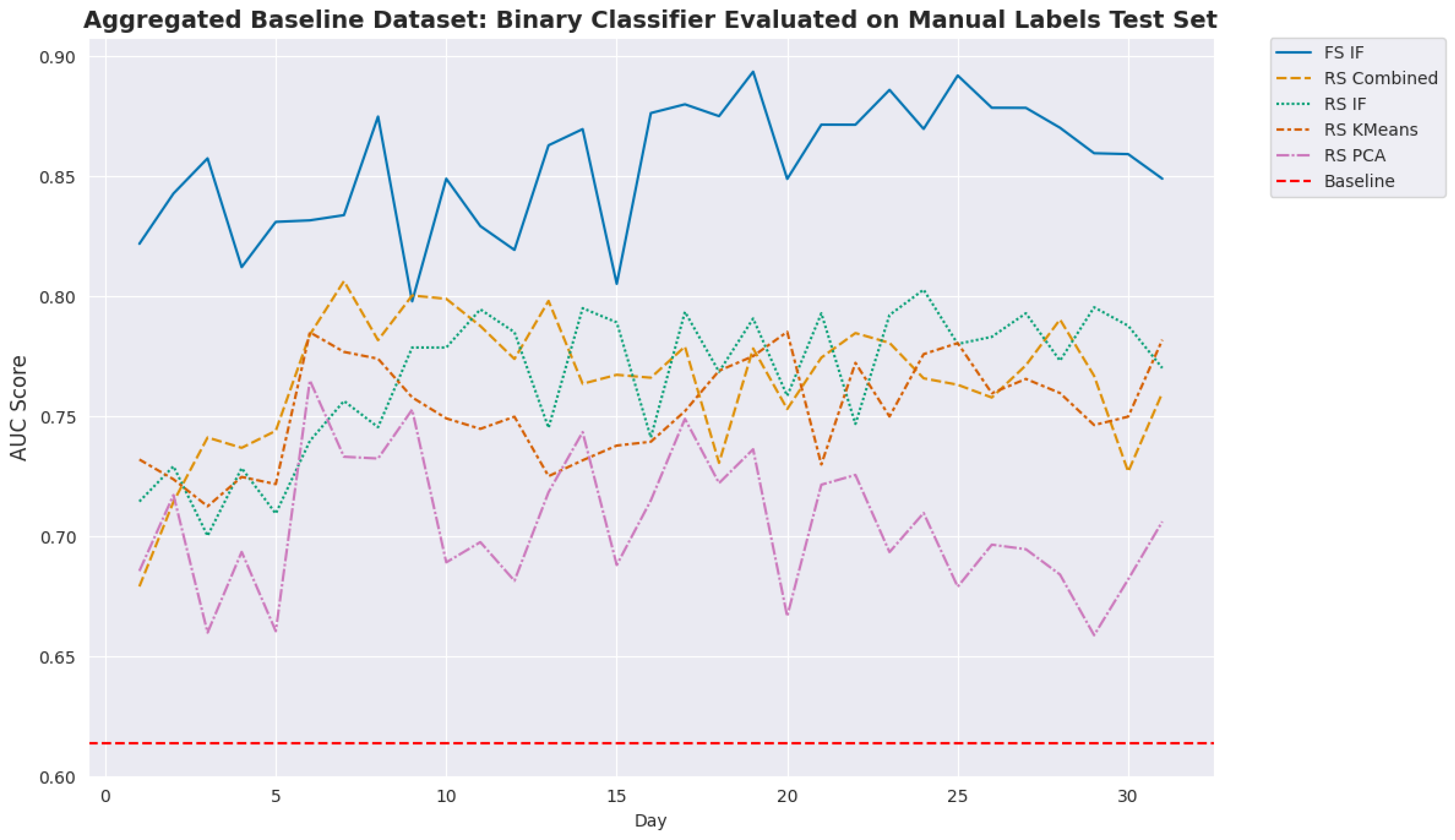
Timesplit

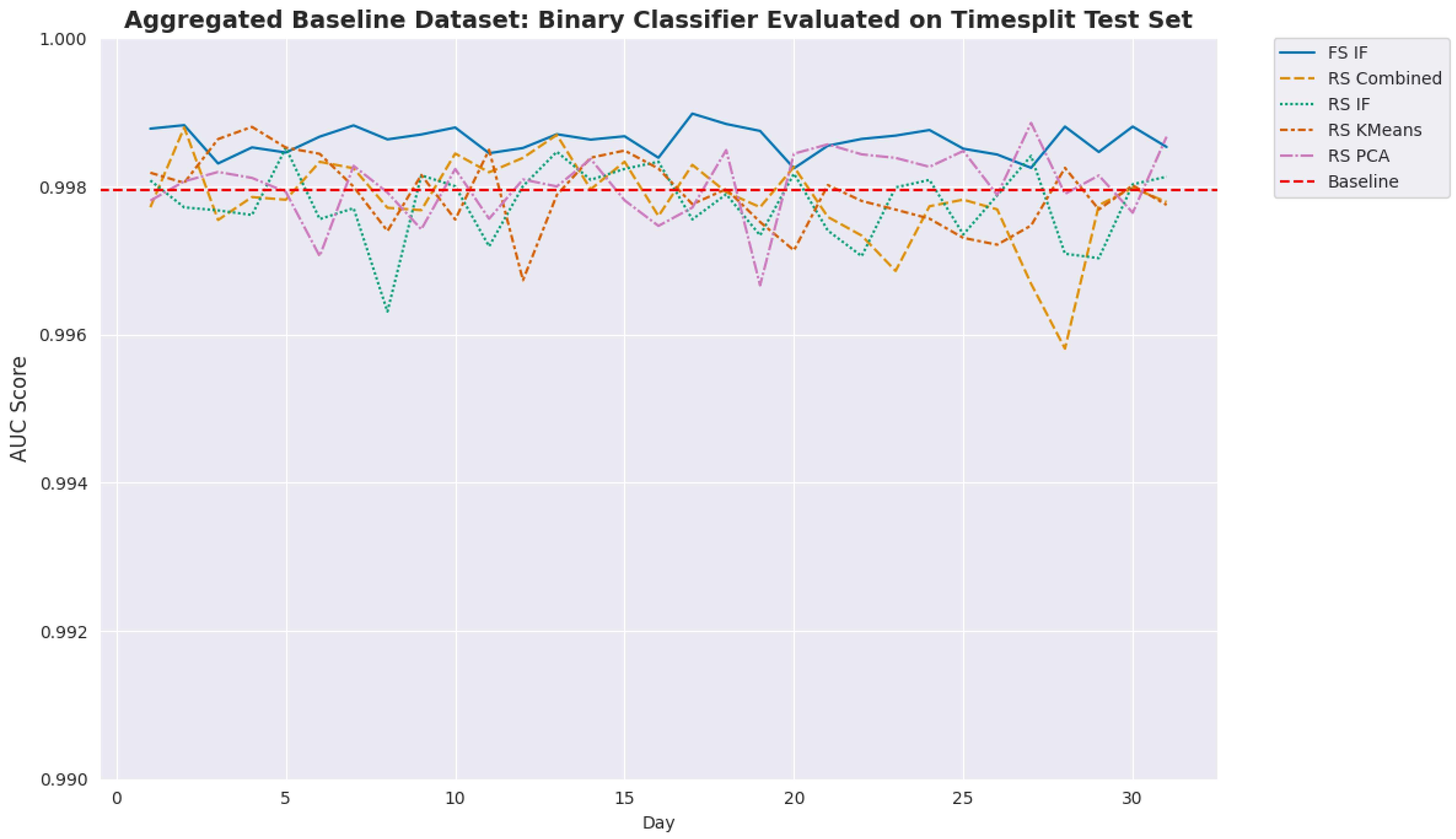
Easier

Manual Labels

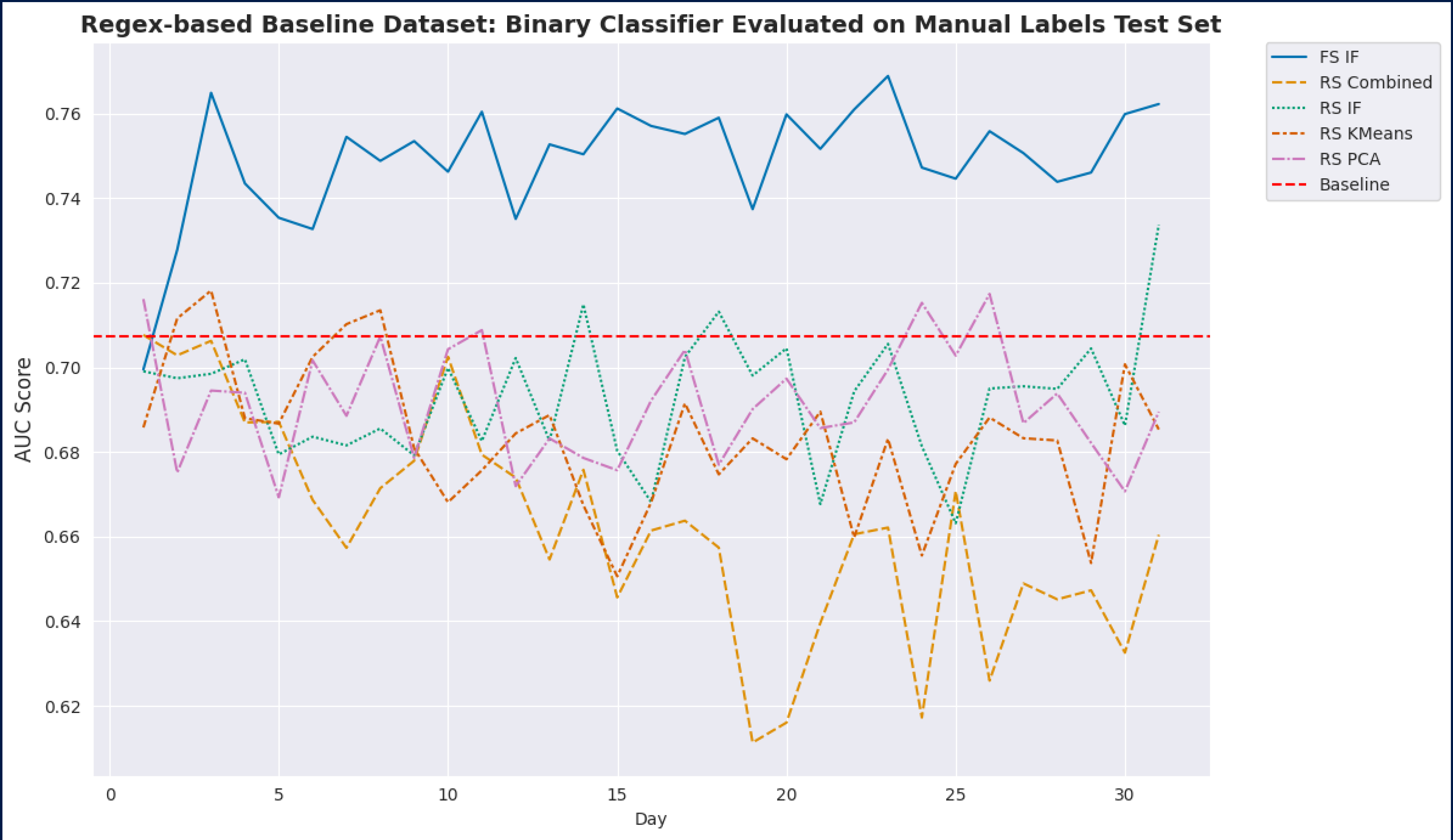
Harder

Training Set	Manual Label AUC	Timesplit Test AUC
Aggregated Baseline (AB)	0.6138	0.9979
AB + Full-Scale	**0.8935**	**0.9990**
AB + Reduced-Scale Combined	0.8063	0.9988
AB + Reduced-Scale IF	0.8028	0.9985
AB + Reduced-Scale KMeans	0.7852	0.9988
AB + Reduced-Scale PCA	0.7650	0.9989
Regex-based Baseline (RB)	0.7072	0.9988
RB + Full-Scale	**0.7689**	0.9990
RB + Reduced-Scale Combined	0.7077	0.9995
RB + Reduced-Scale IF	0.7337	**0.9998**
RB + Reduced-Scale KMeans	0.7182	0.9994
RB + Reduced-Scale PCA	0.7174	0.9996

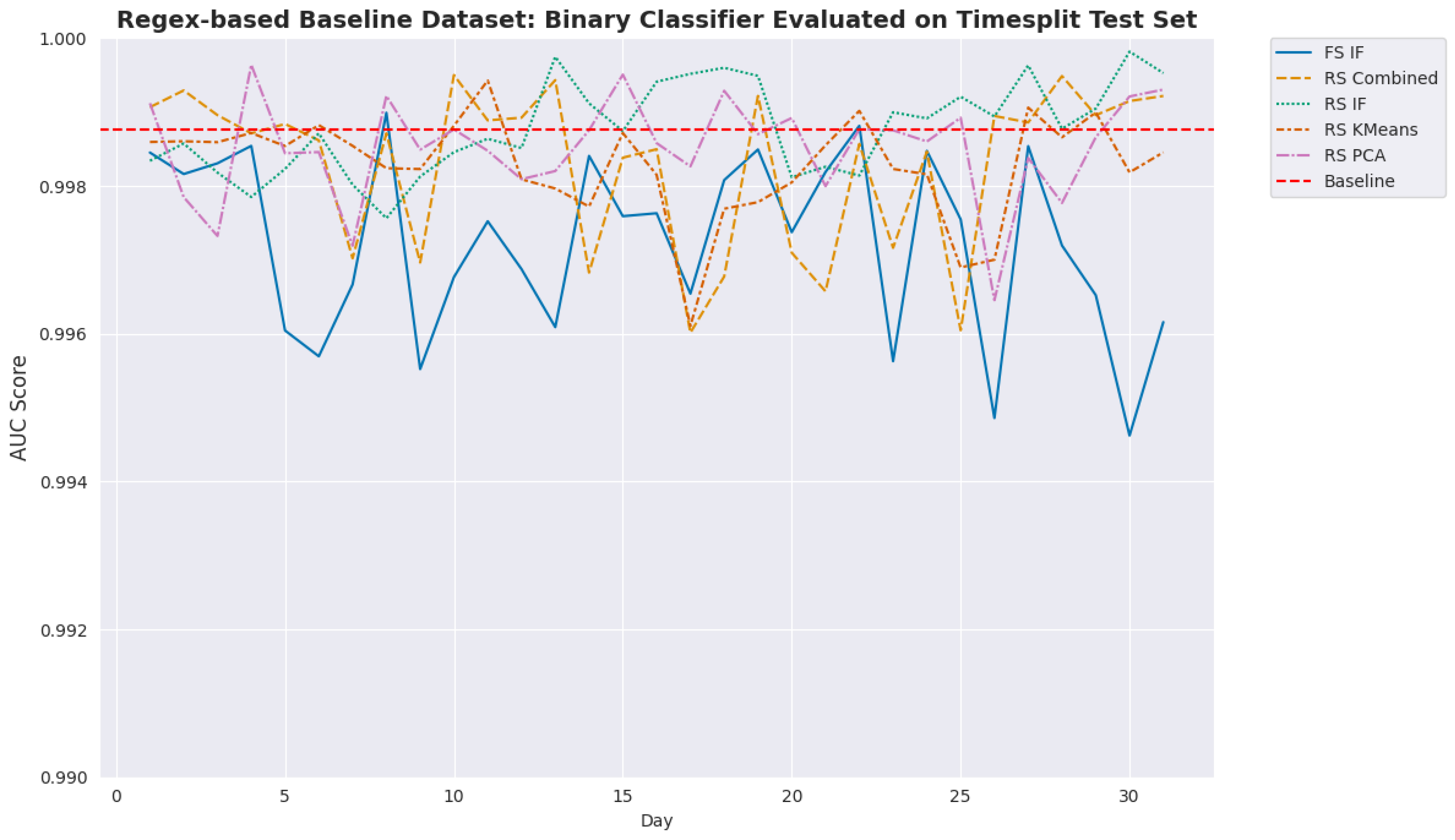

Aggregated Baseline Dataset: Binary Classifier Evaluated on Manual Labels Test Set
AUC Score
Day
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0
5
10
15
20
25
30
FS IF
RS Combined
RS IF
RS KMeans
RS PCA
Baseline



Aggregated Baseline Dataset: Binary Classifier Evaluated on Timesplit Test Set
AUC Score
Day
1.000
0.998
0.996
0.994
0.992
0.990
0
5
10
15
20
25
30
FS IF
RS Combined
RS IF
RS KMeans
RS PCA
Baseline



Regex-based Baseline Dataset: Binary Classifier Evaluated on Manual Labels Test Set
AUC Score
Day
FS IF
RS Combined
RS IF
RS KMeans
RS PCA
Baseline
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0
5
10
15
20
25
30


Regex-based Baseline Dataset: Binary Classifier Evaluated on Timesplit Test Set
AUC Score
Day
1.000
0.998
0.996
0.994
0.992
0.990
0
5
10
15
20
25
30
FS IF
RS Combined
RS IF
RS KMeans
RS PCA
Baseline

Conclusion

Do You Qualify for Benign Anomaly Detection

- Big data?
- New data coming in?
- Cybersecurity machine learning model?

Do You Qualify for Benign Anomaly Detection

Monday Morning

- Pick a cybersecurity model that needs updating
- Dig up some recent data
- Run isolation forest
- Send anomalies to a small reasoning LLM
  - (Confirm benign labels)
- Retrain target model

Black Hat Sound Bytes

- Anomaly detection excels at locating benign data in the long tail
- Modern LLMs have enabled automated pipelines for benign data labeling that were not possible before
- Training set augmentation with benign anomalies is a generalizable method for improving cybersecurity models

Appendix

Expert Features

- Character length
- Proportion of operators:
  - `{'%', '*', '^', '`', '/', '+', '-', '=', '>'}`
- Proportion of upper-case characters
- Proportion of lower-case characters
- ASCII per-character counts
- Shannon entropy

Expert Features cont.

- Count of `"echo"` markers
- Count of `"replace"` markers
- Count of `"#"` markers
- Count of markers:
  - `{" -e ", " -ec ", " -enc ", " -encodedcommand ", "frombase64string("}`
- Count of markers:
  - `{"^", '""', "set", "&&", "&&for", "for %", ";;"}`
- Count of markers:
  - `{"http", "www.", ".com", "html", "tcp", "udp"}`
- Count of markers:
  - `{"lsass", "samsrv", "hklm\\sam", "winlogon", "netlogon", "kerberos.dll", "dump", ".bin", "ntds"}`
- Test for deliberate encoding and encryption
- Check for multiple valid file paths
- Check for remote executable
- Check for exactly one hostname and local file path

Spark ML Features

- Normalized tokens
  - WordPunct tokenize: `"\\w+|[^\\w\\s]+"`
  - Replace numeric digits with *
- Normalized tokens -> TF-IDF
- Normalized tokens -> Compute most common 1024 tokens in vocab -> One-hot encoding

LLM Labeler Prompt

We need to determine whether a command line is benign or malicious.

We have to be very confident in our answer, so let's think about this deeply.

Let's start by explaining what the command line does.

Then explain how, if possible, the command could be used in a benign way.

Then explain how, if possible, the command line could be used maliciously.

Finally, make a verdict on whether the command line is benign or malicious.

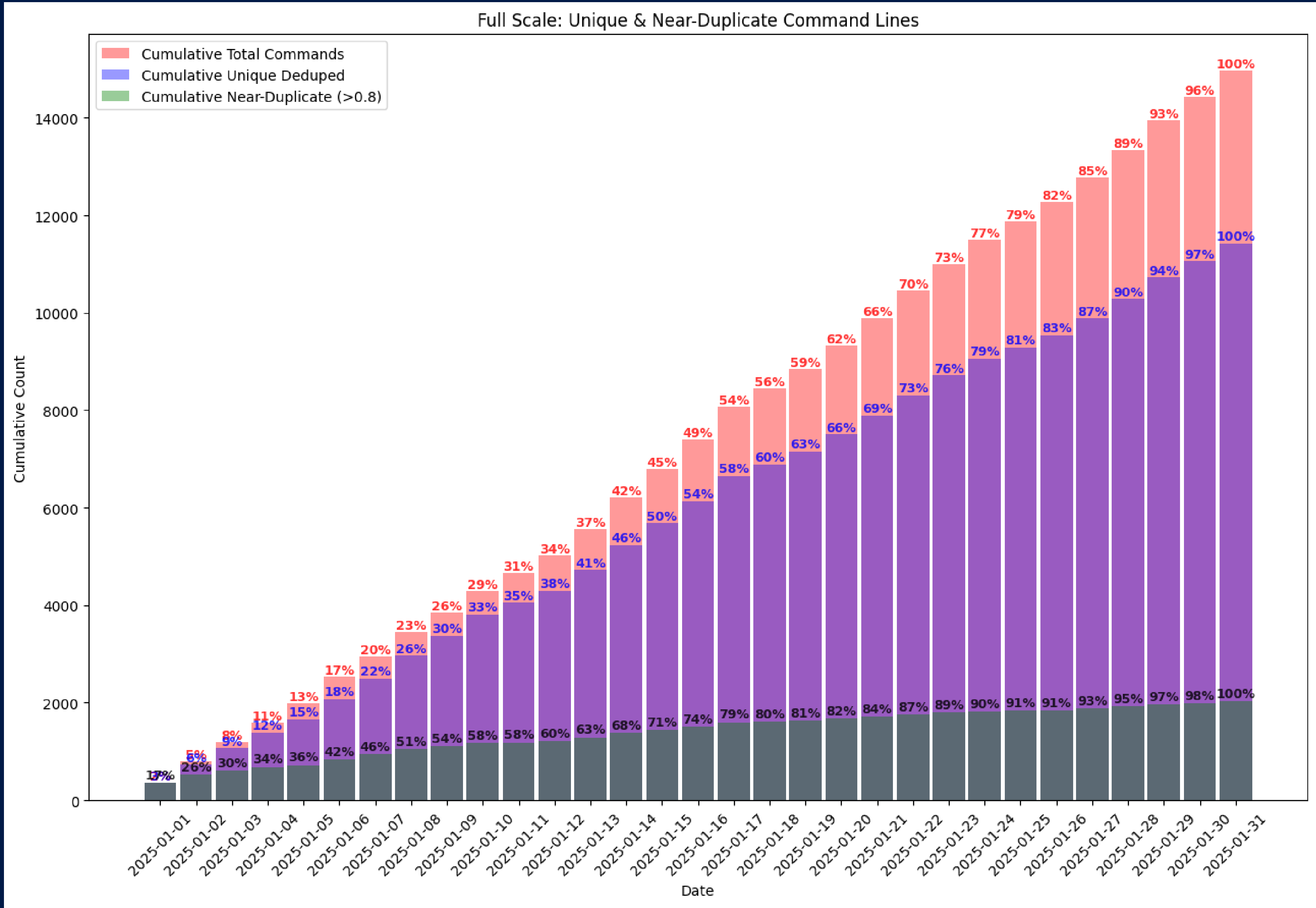
Output your response as either VERDICT_BENIGN or VERDICT_MALICIOUS.

Once again, only convict a command as malicious if it is very likely.

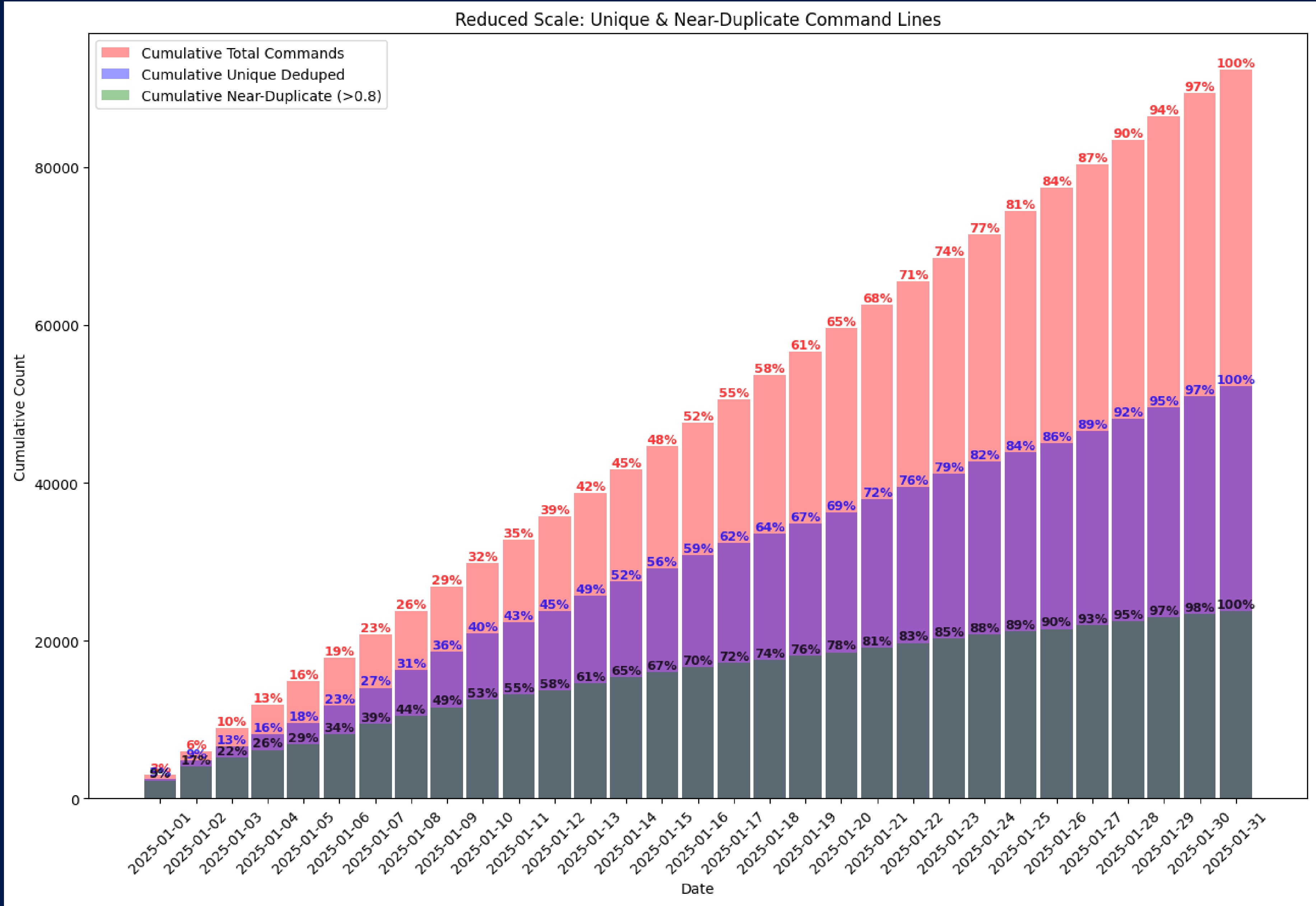
Use the term VERDICT_BENIGN or VERDICT_MALICIOUS only ONCE in your response.

Command line:

Full-scale Command-line Distribution



Reduced-scale Command-line Distribution



References

[1] Vinay, V., & Mangal, A. (2024). SCADE: Scalable Command-line Anomaly Detection Engine. *arXiv preprint arXiv:2412.04259*.

[2] Nisslmueller, U. (2022). *LOLBin detection through unsupervised learning: An approach based on explicit featurization of the command line and parent-child relationships* (Master's thesis, University of Twente).

[3] Filar, B., & French, D. (2020). Problemchild: Discovering anomalous patterns based on parent-child process relationships. *arXiv preprint arXiv:2008.04676*.

[4] Hendler, D., Kels, S., & Rubin, A. (2020, October). Amsi-based detection of malicious powershell code using contextual embeddings. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (pp. 679-693).

[5] Hendler, D., Kels, S., & Rubin, A. (2018, May). Detecting malicious powershell commands using deep neural networks. In *Proceedings of the 2018 on Asia conference on computer and communications security* (pp. 187-197).