CHARLOTTE

THE GRADUATE SCHOOL

ANOMALY DETECTION IN CYBERSECURITY

By -Ashlesha Gupta

CHARLOTTE
THE GRADUATE SCHOOL

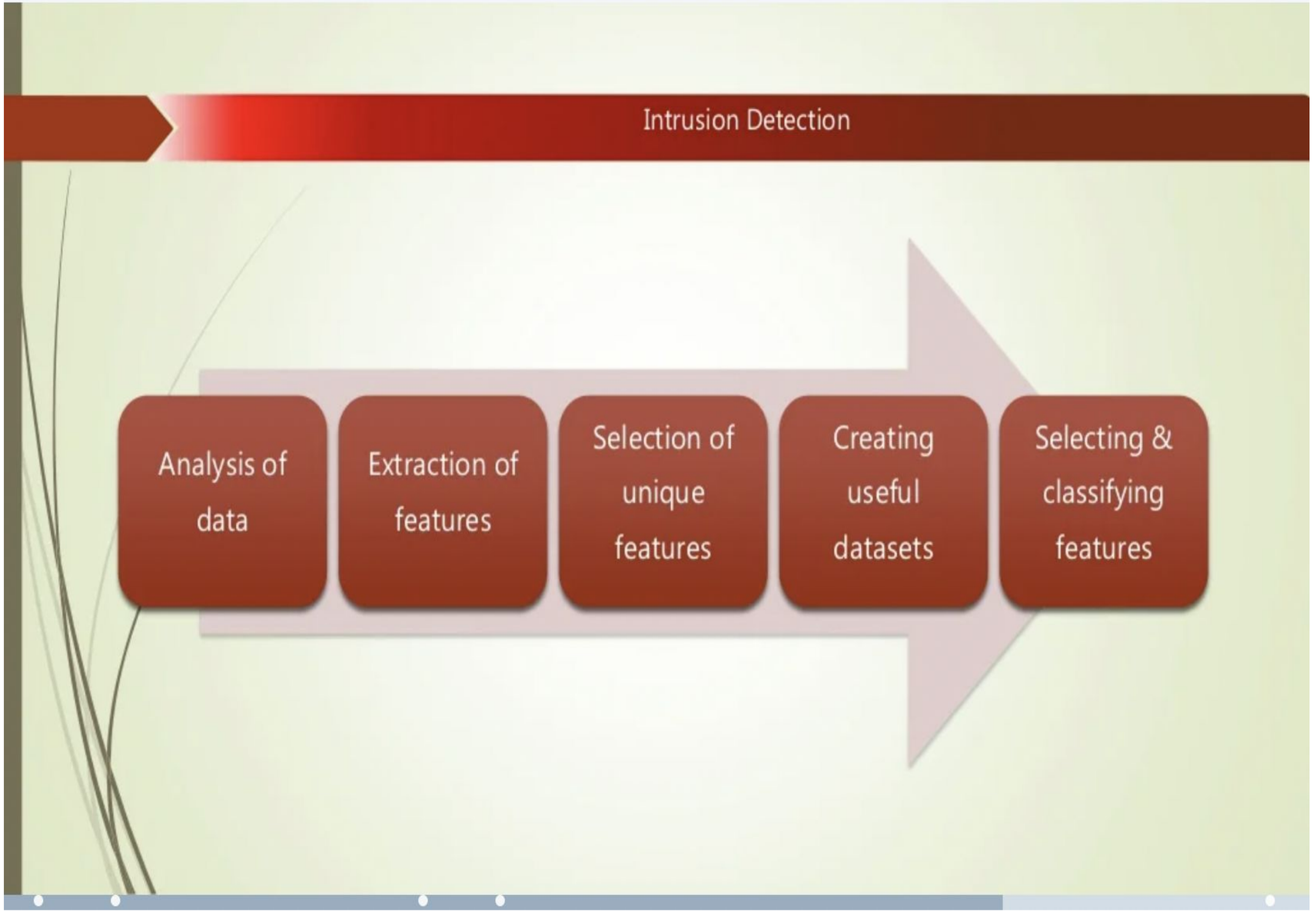
CHARLOTTE
THE GRADUATE SCHOOL

PROBLEM

- Cybersecurity attacks are growing both in frequency and sophistication over the years.
- The development of effective techniques has been an urgent demand in the field of the cyber security community.
- Machine learning can be leveraged to detect the cybersecurity breaches and attacks.
- This project will explore Machine Learning as a viable solution by examining its capabilities to classify malicious traffic in a network and help identify potential cyber threats and attacks .

CHALLENGES


Cyber Security Is Failing
We have to ask why...
• Are attackers improving?
• Are businesses getting worse (at protecting data)?
It seems the likelihood of a breach is increasing....
HaveIBeenPwned.com now holds >5b
accounts.
Review your spend to minimise risk.

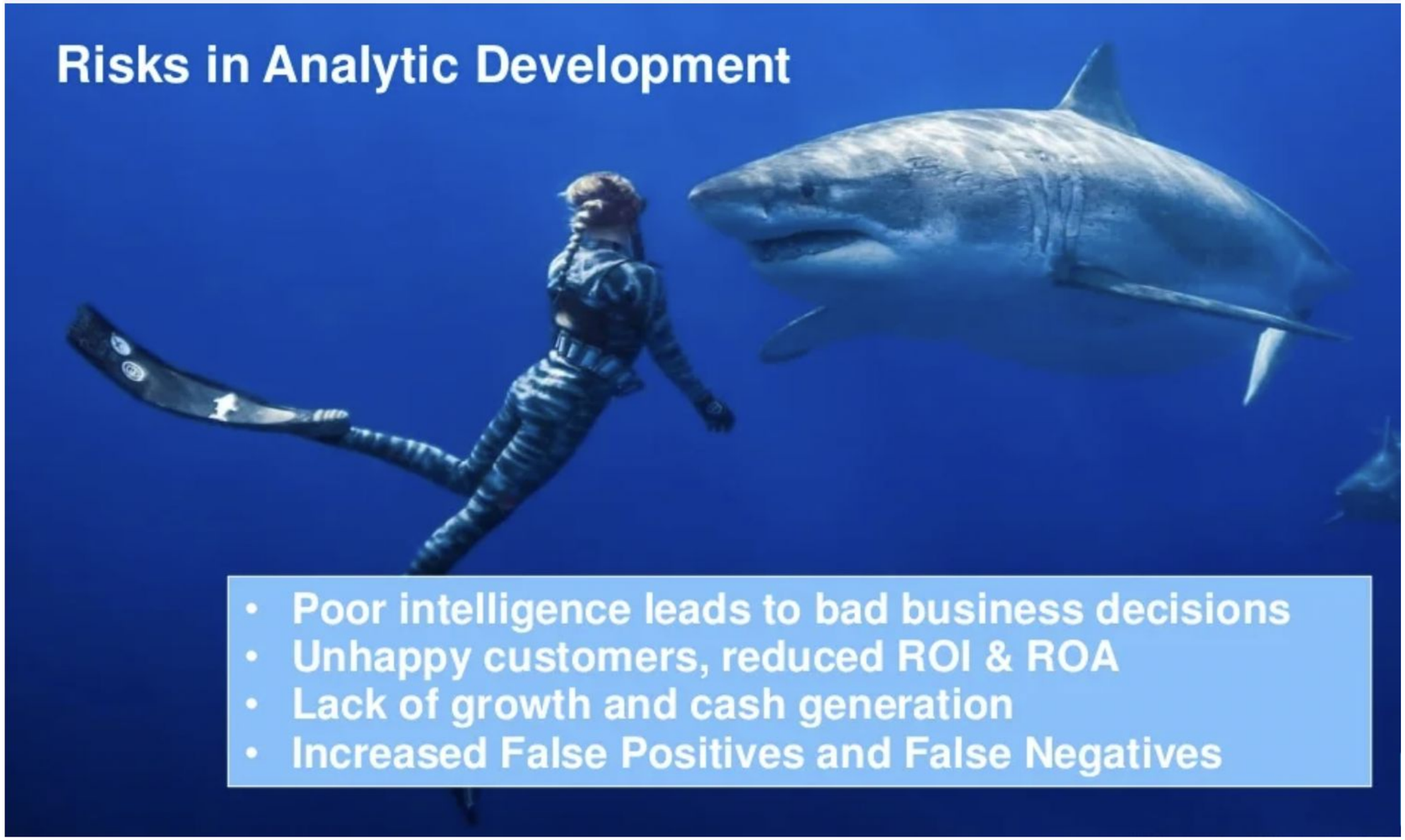


CHARLOTTE
THE GRADUATE SCHOOL

CHALLENGES

- Due to shortage of skilled professionals it has become very difficult for the cyber security professionals to analyse millions of low fidelity alerts manually .
- The security landscape is changing rapidly.
- Attackers have started using advanced technology and cybersecurity analysts are struggling to get to them.

CHALLENGES


Risks in Analytic Development
• Poor intelligence leads to bad business decisions
• Unhappy customers, reduced ROI & ROA
• Lack of growth and cash generation
• Increased False Positives and False Negatives

Motivation

Having worked in the cyber security industry , I myself have faced the challenge of juggling among various websites to analyse the network traffic and then finally finding them to be false-positives.Brainstorming on alerts that are not contributing to the attack is really time consuming and hectic. And that's how I decided to tackle this issue using machine learning.

ANOMALY DETECTION

- Anomaly detection is the process of finding the outliers in the data, i.e. points that are significantly different from the majority of the other data points.
- An outlier is nothing but a data point that differs significantly from other data points in the given dataset.

ANOMALY DETECTION USING ISOLATION FORREST

- Isolation forest is a machine learning algorithm for anomaly detection.
- It's an unsupervised learning algorithm that identifies anomaly by isolating outliers in the data.

Existing Related Approaches

- 1. When machine learning was not so popular , Cybersecurity analysts used to automate their low fidelity alerts analysis by writing a core python code and using the APIs of those tools. But a lot of tools required proprietary access to the APIs and code(SET OF RULES) used to be written by Analysts , so again not an effective way as we can leverage machine learning now.

Existing Related Approaches

- **Network Based Intrusion Detection**
  Two stacking models are developed by combining ensemble algorithms with simple supervised algorithms KNN and NN respectively. We have used the UNSW-NB15 dataset which has 175, 341 train and 82,332 test data. In this paper, a network intrusion detection system built with the stacking ML model is proposed where XGBoost and KNN act as base classifiers and RF as a meta classifier.

- **Network Based Intrusion Detection**

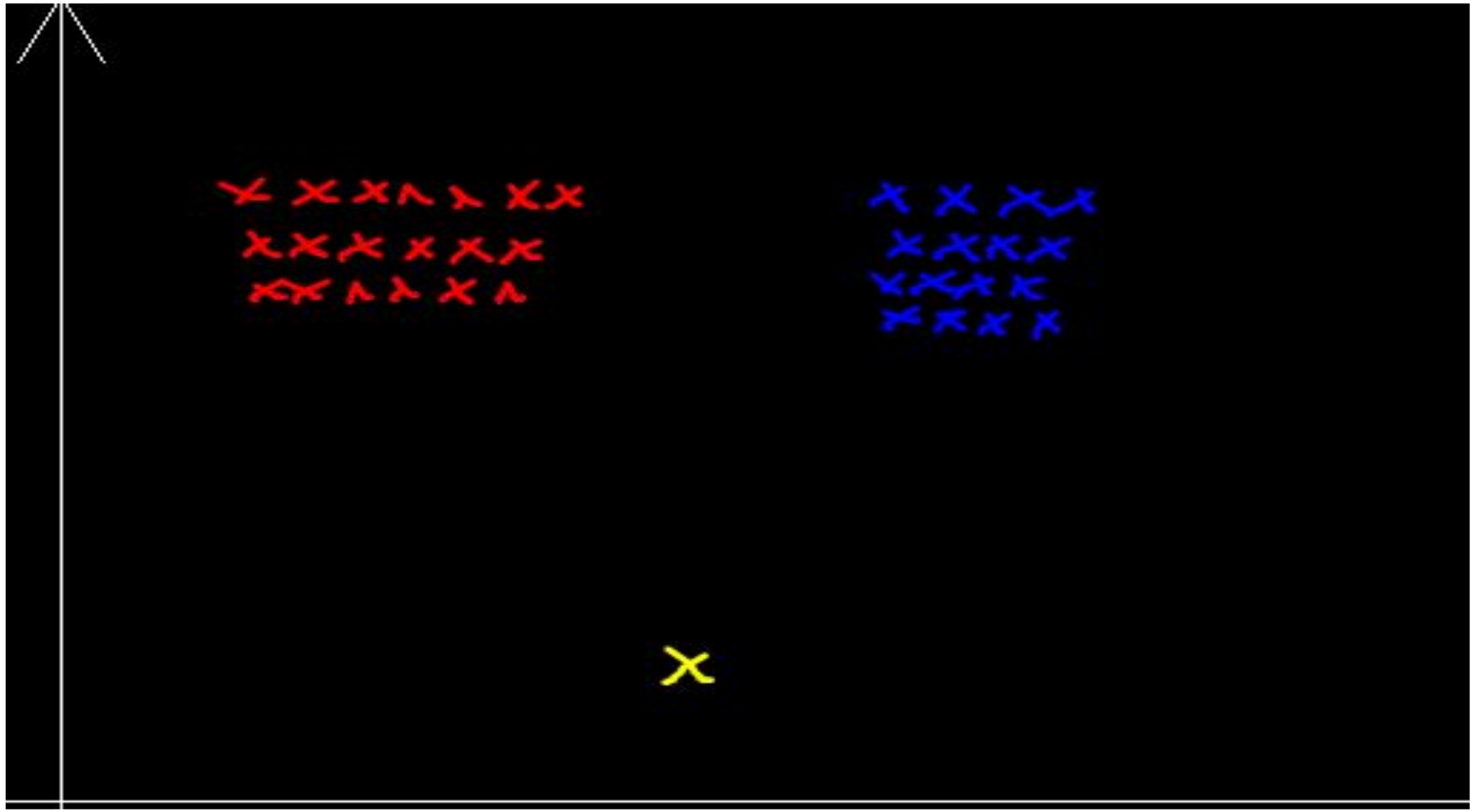
Citation: Tufan, Emrah & Tezcan, Cihangir & Acarturk, Cengiz. (2021). Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network. IEEE Access. 9. 50078-50092. 10.1109/ACCESS.2021.3068961.

Date Published : 26 March 2021

Location : IEEE Access conference.

Authors:Emrah Tufan,Cihangir Tezcan,Cengiz Acartürk

Limitations Of K-Nearest Neighbour



UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

Paper 1 : Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network

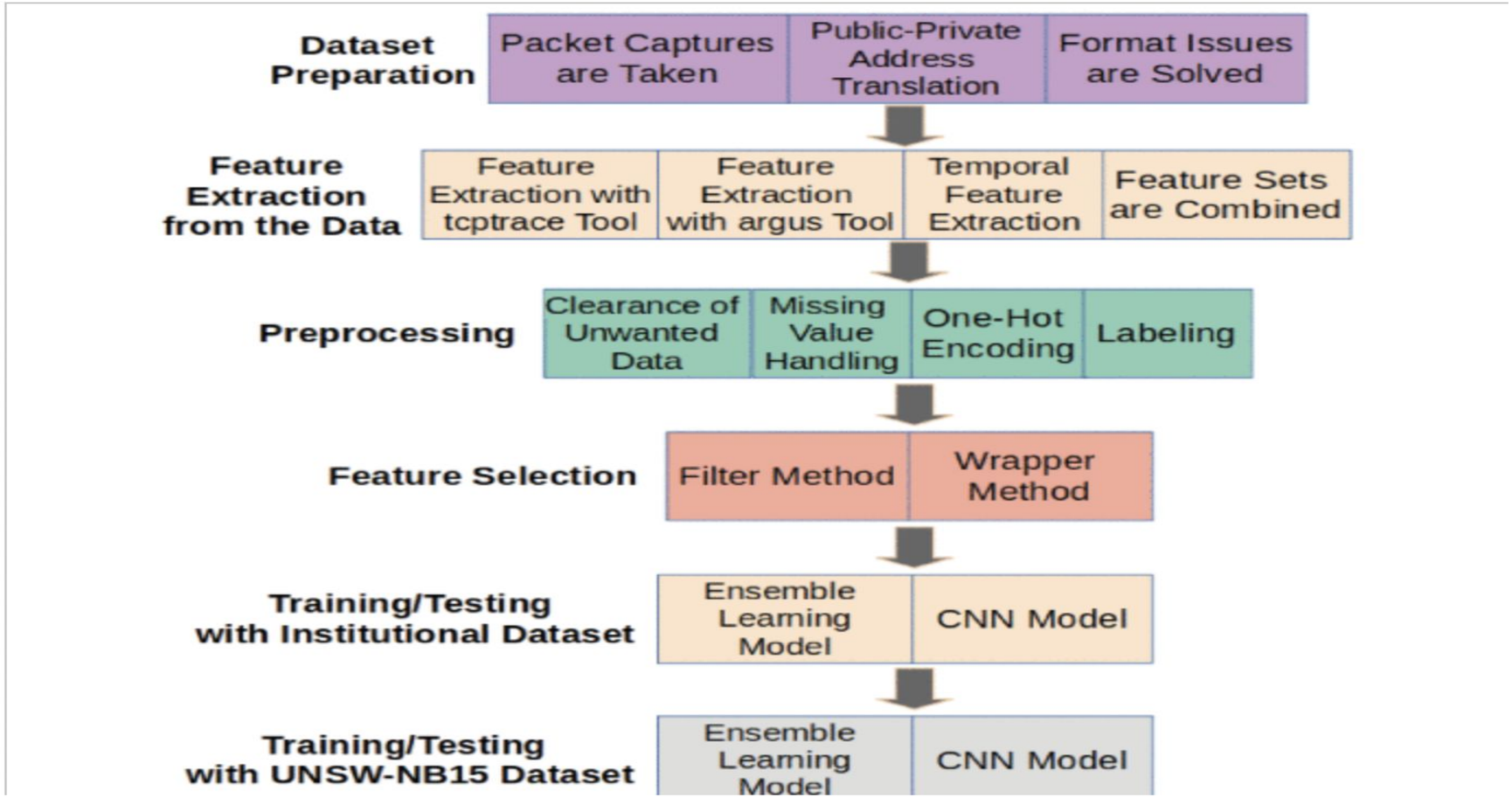
Citation: Tufan, Emrah & Tezcan, Cihangir & Acarturk, Cengiz. (2021). Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network. IEEE Access. 9. 50078-50092. 10.1109/ACCESS.2021.3068961.

Date Published : 26 March 2021

Location : IEEE Access conference.

Authors:Emrah Tufan,Cihangir Tezcan,Cengiz Acartürk

Paper 1 : Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network



UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

Paper 1 : Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network

Why I chose this paper ? How exactly does the selected paper solve the problem?

I chose this paper because it gives a clear comparison of all the models (SVM,LR,KNN,CNN,Random Forest) etc. for evaluating an efficient Anomaly detection Algorithm and Intrusion Detection follows anomaly detection which I am going to work on. And it also compared the approaches followed in Rule based Intrusion Detection and Anomaly based intrusion detection. After comparing the efficiency of Models I was introduced to the new concept white box and black box methods for cybersecurity and this research paper will definitely complement the findings of my project .

The selected paper solves the problem by following the complete procedures like:

1.Data set Preparation.

2.Feature Extraction from the data

3.Preprocessing

4.Feature Selection

5.Training /Testing with both the data sets to get the efficiency and accuracy

Existing Related Approaches

Cyber Attack Detection thanks to Machine Learning Algorithms

a strong data analysis is performed resulting in 22 extracted features from the initial Netflow datasets. All these features are then compared with one another through a feature selection process. Then, our approach analyzes five different machine learning algorithms against NetFlow dataset containing common botnets. The Random Forest Classifier succeeds in detecting more than 95% of the botnets in 8 out of 13 scenarios and more than 55% in the most difficult datasets. Finally, insight is given to improve and generalize the results, especially through a bootstrapping technique.

Citation:
https://doi.org/10.48550/arXiv.2001.06309

Explaining The BETH DataSet In Brief

The BETH dataset now contains 8,004,918 events collected from 23 honeypots over the course of around five hours on a major cloud provider.BETH includes modern host activity as well as data from cloud services, making it useful in the world.

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

Explaining The BETH DataSet In Brief

BETH Dataset: Real Cybersecurity Data for Anomaly Detection Research

Table 2. The description and type of each feature within the kernel-level process logs, tracking every create, clone, and kill process call. Starred features were included in the model baselines and converted as described in Appendix A.

FEATURE	TYPE	DESCRIPTION
TIMESTAMP	FLOAT	SECONDS SINCE SYSTEM BOOT
PROCESSID*	INT	INTEGER LABEL FOR THE PROCESS SPAWNING THIS LOG
THREADID	INT	INTEGER LABEL FOR THE THREAD SPAWNING THIS LOG
PARENTPROCESSID*	INT	PARENT'S INTEGER LABEL FOR THE PROCESS SPAWNING THIS LOG
USERID*	INT	LOGIN INTEGER ID OF USER SPAWNING THIS LOG
MOUNTNAMESPACE*	INT (LONG)	SET MOUNTING RESTRICTIONS THIS PROCESS LOG WORKS WITHIN
PROCESSNAME	STRING	STRING COMMAND EXECUTED
HOSTNAME	STRING	NAME OF HOST SERVER
EVENTID*	INT	ID FOR THE EVENT GENERATING THIS LOG
EVENTNAME	STRING	NAME OF THE EVENT GENERATING THIS LOG
ARGSNUM*	INT	LENGTH OF ARGS
RETURNVALUE*	INT	VALUE RETURNED FROM THIS EVENT LOG (USUALLY 0)
STACKADDRESSES	LIST OF INT	MEMORY VALUES RELEVANT TO THE PROCESS
ARGS	LIST OF DICTIONARIES	LIST OF ARGUMENTS PASSED TO THIS PROCESS
SUS	INT (0 OR 1)	BINARY LABEL AS A SUSPICIOUS EVENT (1 IS SUSPICIOUS, 0 IS NOT)
EVIL	INT (0 OR 1)	BINARY AS A KNOWN MALICIOUS EVENT (0 IS BENIGN, 1 IS NOT)

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

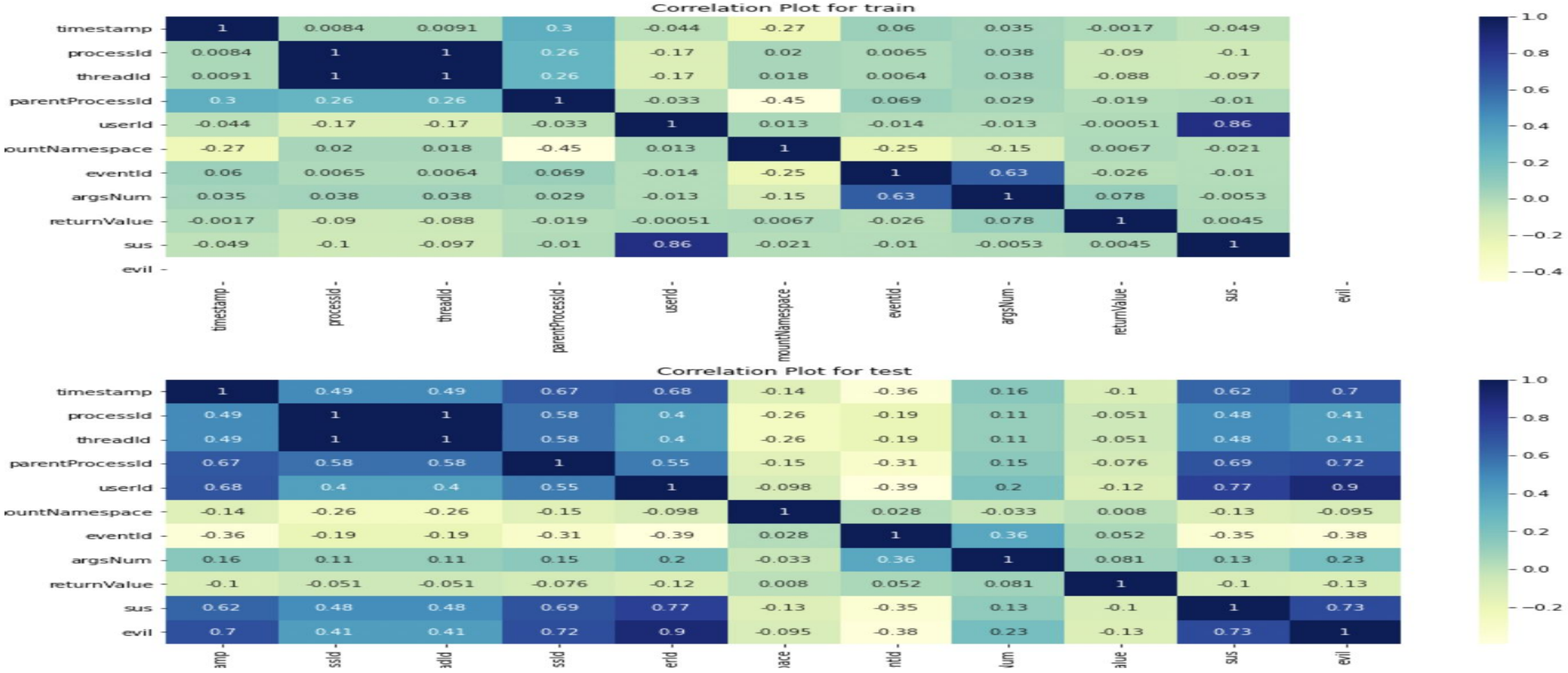
Explaining The BETH DataSet In Brief

BETH Dataset: Real Cybersecurity Data for Anomaly Detection Research

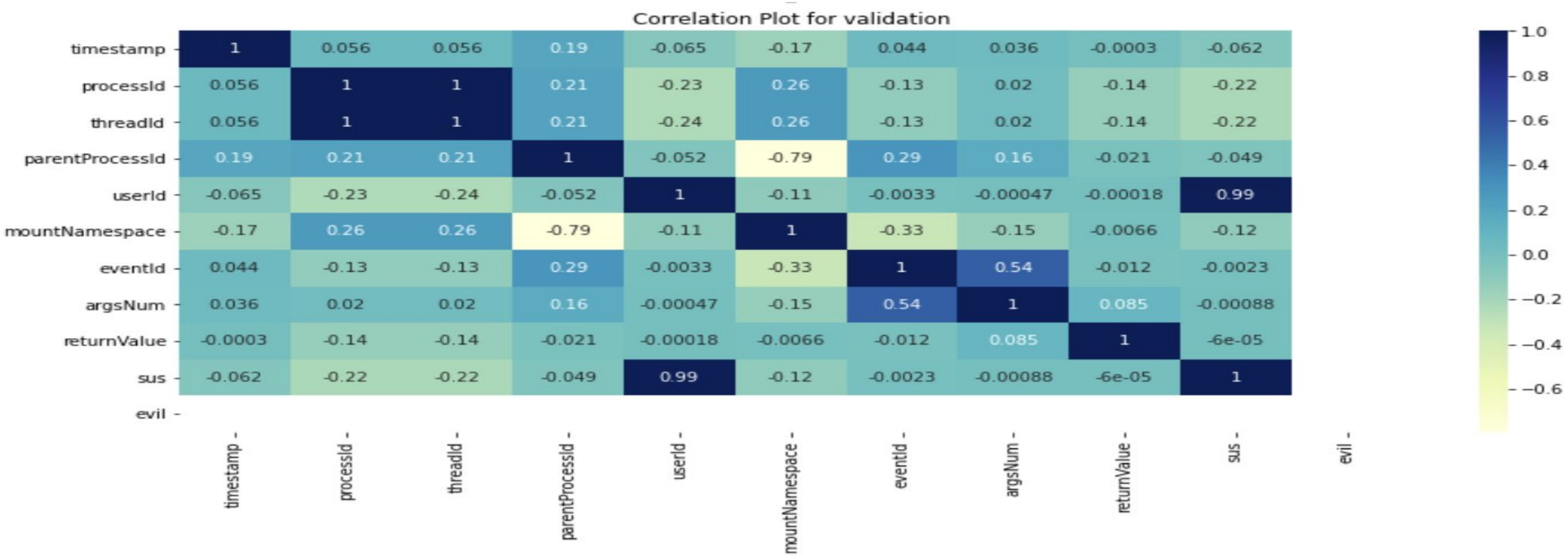
Table 2. The description and type of each feature within the kernel-level process logs, tracking every create, clone, and kill process call. Starred features were included in the model baselines and converted as described in Appendix A.

FEATURE	TYPE	DESCRIPTION
TIMESTAMP	FLOAT	SECONDS SINCE SYSTEM BOOT
PROCESSID*	INT	INTEGER LABEL FOR THE PROCESS SPAWNING THIS LOG
THREADID	INT	INTEGER LABEL FOR THE THREAD SPAWNING THIS LOG
PARENTPROCESSID*	INT	PARENT'S INTEGER LABEL FOR THE PROCESS SPAWNING THIS LOG
USERID*	INT	LOGIN INTEGER ID OF USER SPAWNING THIS LOG
MOUNTNAMESPACE*	INT (LONG)	SET MOUNTING RESTRICTIONS THIS PROCESS LOG WORKS WITHIN
PROCESSNAME	STRING	STRING COMMAND EXECUTED
HOSTNAME	STRING	NAME OF HOST SERVER
EVENTID*	INT	ID FOR THE EVENT GENERATING THIS LOG
EVENTNAME	STRING	NAME OF THE EVENT GENERATING THIS LOG
ARGSNUM*	INT	LENGTH OF ARGS
RETURNVALUE*	INT	VALUE RETURNED FROM THIS EVENT LOG (USUALLY 0)
STACKADDRESSES	LIST OF INT	MEMORY VALUES RELEVANT TO THE PROCESS
ARGS	LIST OF DICTIONARIES	LIST OF ARGUMENTS PASSED TO THIS PROCESS
SUS	INT (0 OR 1)	BINARY LABEL AS A SUSPICIOUS EVENT (1 IS SUSPICIOUS, 0 IS NOT)
EVIL	INT (0 OR 1)	BINARY AS A KNOWN MALICIOUS EVENT (0 IS BENIGN, 1 IS NOT)

Explaining The BETH DataSet In Brief



Explaining The BETH DataSet In Brief



Explaining The BETH DataSet In Brief

- All three of the datasets have a heavy correlation between userid and the associated labels which feature in the dataset (sus and/or evil).
- processid and threadid are highly correlated and seem to have similar correlation values across all three datasets. This means that they are representing pretty much the same thing and one of them could be dropped.
- The correlation plots all look significantly different. This probably means it’s a hard problem!

The METHOD

As discussed in the previous slides , I have implemented ISOLATION FOREST Algorithm which is also called as Anomaly Detection Algorithm and Random Forest Algorithm on the BETH data set to detect the anomaly in the BETH dataset

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

My Replication With The help of Kaggle's BETH dataset

As discussed in the previous slides , I have implemented ISOLATION FOREST Algorithm which is also called as Anomaly Detection Algorithm and Random Forest Algorithm on the BETH data set to detect the anomaly in the BETH dataset

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

ISOLATION FOREST

In [70]:
```
clf = IsolationForest(contamination=0.1, random_state=0).fit(train_df_feats)
```

In [71]:
```
y_pred= clf.predict(val_df_feats)
y_probas = clf.score_samples(val_df_feats)
metric_printer(val_df_labels, y_pred)
```

```
Precision:      0.9923146576418195
Recall:         0.460
F1-Score:       0.626
```

In [72]:
```
y_pred= clf.predict(test_df_feats)
y_probas = clf.score_samples(test_df_feats)
metric_printer(test_df_labels, y_pred)
```

```
Precision:      0.8951880300409012
Recall:         0.893
F1-Score:       0.894
```

In [73]:
```
train_non_outliers = train_df_feats[train_df_labels==0]
clf = linear_model.SGDOneClassSVM(random_state=0).fit(train_non_outliers)
```

In [74]:
```
y_preds = clf.predict(val_df_feats)
metric_printer(val_df_labels, y_preds)
```

```
Precision:      1.0
Recall:         0.177
F1-Score:       0.300
```

In [75]:
```
y_preds = clf.predict(test_df_feats)
metric_printer(test_df_labels, y_preds)
```

```
Precision:      1.0
Recall:         0.260
F1-Score:       0.412
```

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

RANDOMFOREST

```
clf = RandomForestClassifier(max_depth=2, random_state=0).fit(test_df_feats, test_df_labels)
y_preds = clf.predict(val_df_feats)
metric_printer(val_df_labels, y_preds)
```

```
Precision:      1.0
Recall:         1.000
F1-Score:       1.000
```

In [82]:
```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(max_depth=2, random_state=0).fit(val_df_feats, val_df_labels)
y_preds = clf.predict(test_df_feats)
metric_printer(test_df_labels, y_preds)
```

```
Precision:      1.0
Recall:         1.000
F1-Score:       1.000
```

UNIVERSITY OF NORTH CAROLINA CHARLOTTE | THE GRADUATE SCHOOL

SUPPORT VECTOR MACHINE

In [75]:
```
train_non_outliers = train_df_feats[train_df_labels==0]
clf = linear_model.SGDOneClassSVM(random_state=0).fit(train_non_outliers)
```

In [76]:
```
y_preds = clf.predict(val_df_feats)
metric_printer(val_df_labels, y_preds)
```

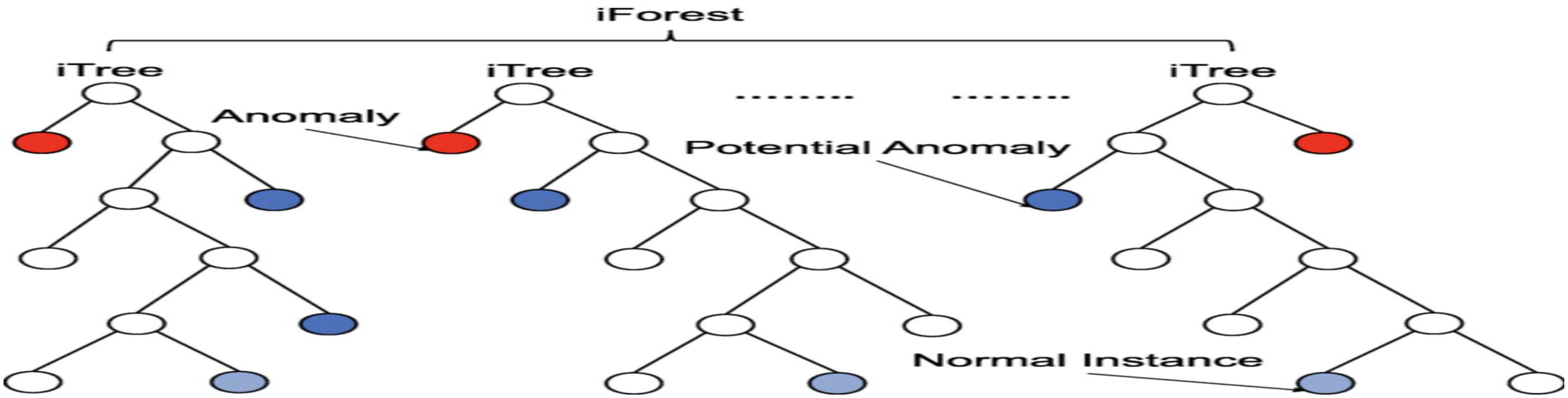
```
Precision:      1.0
Recall:         0.177
F1-Score:       0.300
```

In [77]:
```
y_preds = clf.predict(test_df_feats)
metric_printer(test_df_labels, y_preds)
```

```
Precision:      1.0
Recall:         0.260
F1-Score:       0.412
```

PERFORMANCE

Out of ISOLATION FOREST , RANDOM FOREST And SUPPORT VECTOR MACHINE . RANDOM FOREST performed well , but ISOLATION FOREST performed consistently on all 3 datasets .



Isolation Forest: learned iForest construction for toy dataset