

North Carolina Agricultural and Technical State University

MACHINE LEARNING BASED IOT INTRUSION DETECTION USING THE MQTT-IOT-IDS2020 & IOTIDS20 DATASETS

Steve Chesney, PhD Candidate, MBA
Member of the Center for Cyber Defense
North Carolina A&T State University
Advisor: Dr. Kaushik Roy

AGGIES DO

Agenda

- Introduction
- The MQTT-IoT-IDS2020 Overview
  - Dataset Description
  - Feature Selection
  - Methodology
  - Results
  - Future Work
- The IoT Network Intrusion Dataset Overview
  - Data Collection and Preprocessing Steps
  - Data Modeling & Results
  - Future Work
- Acknowledgements
- References

Introduction

The Internet of Things

"A global infrastructure for the information society, enabling advanced services by interconnecting things (physical and virtual) based on existing and evolving interoperable information and communication technologies



- Sensors
- Actuators
- Smartphones
- Smart-watches
- Wearable devices
- Medical devices
- RFID-tagged animals
- Smart-appliances
  - Refrigerator
  - Washing machine
  - Dish washer
  - Coffee machine
- Automobile components
- IIoT
- MIoT, IoBT

ITU, 2018

IoT Device Explosion & Big Data

- The number of IoT devices has grown rapidly - both household and industrial (IIoT). By 2018 there were over 23 billion of IoT devices were connected to Internet
- This growth has brought about an enormous amount data and network traffic, which can be classified as Big Data



Lucero, 2016

The Dimensions of IoT Devices

- Day
- Night
- On the move

ANY TIME communication

- At the computer
- Outdoor
- Indoor (away from the computer)

ANY PLACE communication

- Between computers
- Thing to thing
- Human to thing
- Human to human

ANY THING communication

Doffman, 2019

Problem Statement



- Traditional methods for cyber-security are obsolete
- IoT devices are heterogeneous
  - Different types
  - Different methods of communication (protocols)
  - Different types of data being shared
- IoT devices are too numerous - billions
- Edge devices (not protected by central network) are vulnerable
- IoT devices have a limited compute resources (constrained devices):
  - Processing
  - Memory
  - Storage
  - Networking

Doffman, 2019

Cyber Security of IoT: Why it Matters…

- **2015** - Average annual loss of $7.7 million per org. - 19% incr.
- **2015** - $500 billion - yearly global loss in
- **2016 to 2017** - 22.7 percentage increase in cybersecurity costs
- **2017 to 2018** - 85% incr. (at peak) in month-to-month cyber attacks
- **2019** - $3.92 million – average total cost of a data breach
- **Malicious attacks** - most common & most expensive - 42% in 2014 to 51% in 2019.
- **Impacts of cyber-attacks:** loss of revenue, loss of privacy, loss of reputation, loss of operation, loss of life (military)

(Greenbard, 2016)

- Traditional methods for cyber-security are obsolete/ineffective for IoT devices

- This presentation gives an overview of the Machine Learning and Deep learning analysis of 2 IoT datasets for the purposes creating effective intrusion detection capabilities to secure IoT device

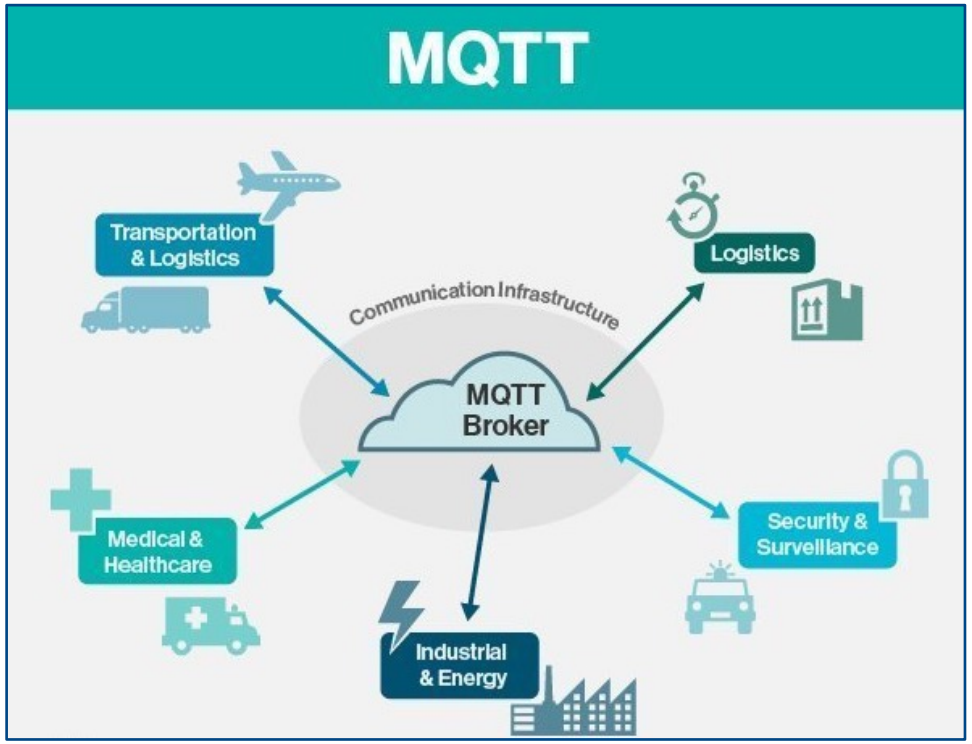
- **IoT Datasets:**
  - MQTT-IoT-IDS2020 (current work)
  - IOTIDS20 (previous work)



MQTT Overview

MQTT Overview

- MQTT is an IoT communications protocol that is used for machine-to-machine communications, created by IBM in 1993
- MQTT is a standardized, light-weight session layer communications protocol with publish/subscribe/broker architecture use for messaging and information exchange among IoT devices
- **Protocol Details:**
  - Can only transmit 256 MB of Data per message (low latency)
  - Messages are assigned a QoS level, which guarantees the delivery of messages between sender and receiver:
    - Level-0 – at most once
    - Level-1 – at least once
    - Level-2 – exactly once
  - An MQTT session is divided into four stages: connection, authentication, communication and termination.



As of 2018, there were 53,396 publicly available and accessible MQTT devices

Harsha, Bhavani & Kundhavai, 2018, pp. 1-2

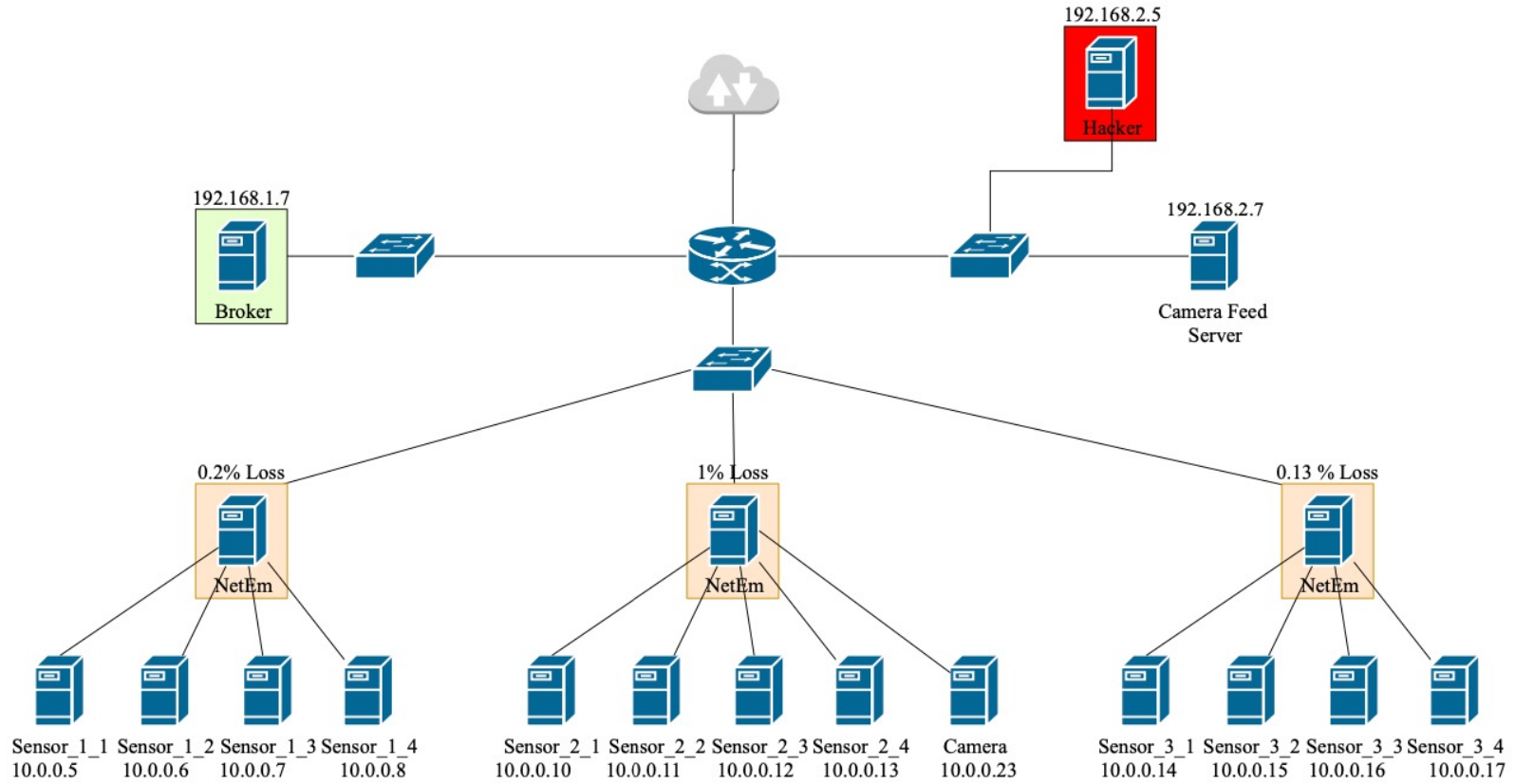
Dataset Description

Dataset Description

- Although widely used, there are security risks associated with MQTT
- Publicly available MQTT datasets are limited for IDS analysis
- The MQTT IDS dataset simulates a realistic MQTT IoT network in a normal operation scenario.
- The dataset includes both generic networking scanning attacks, as well as MQTT brute-force attack.
- The dataset consists of five recorded scenarios; normal operation and four attack scenarios:
  - Aggressive scan (Scan A)
  - User Datagram Protocol (UDP) scan (Scan sU)
  - Sparta SSH brute-force (Sparta)
  - MQTT brute-force attack (MQTT BF)

Hindy, Bures, Tachtatzis & Bellekens, 2020, pp. 2-3

MQTT Network Architecture



Hindy, Bures, Tachtatzis & Bellekens, 2020, p. 3

MQTT IDS Dataset Details & Features

- MQTT brute-force attack (MQTT BF)
  - # of Rows – 10,045,316
  - # of Columns - 31
- Sparta SSH brute-force (Sparta)
  - # of Rows - 20,676,140
  - # of Columns - 31
- Aggressive scan (Scan A)
  - # of Rows – 111,392
  - # of Columns - 31
- User Datagram Protocol (UDP) scan (Scan sU)
  - # of Rows – 233,255
  - # of Columns - 31
- The "**is_attack**" column is the target column used for binary classification in this work

MQTT IDS Dataset Details & Features, cont'd

Feature	Data Type	Description	Packet	Uni-flow	Bi-flow
ip_src	Text	Source IP Address	✓	✓	✓
ip_dest	Text	Destination IP Address	✓	✓	✓
protocol	Text	Last layer protocol	✓		
ttl	Integer	Time to live	✓		
ip_len	Integer	Packet Length	✓		
ip_flag_df	Binary	Don't fragment IP flag	✓		
ip_flag_mf	Binary	More fragments IP flag	✓		
ip_flag_rb	Binary	Reserved IP flag	✓		
prt_src	Integer	Source Port	✓	✓	✓
prt_dst	Integer	Destination Port	✓	✓	✓
proto	Integer	Transport Layer protocol (TCP/UDP)		✓	✓
tcp_flag_res	Binary	Reserved TCP flag	✓		
tcp_flag_ns	Binary	Nonce sum TCP flag	✓		
tcp_flag_cwr	Binary	Congestion Window Reduced TCP flag	✓		
tcp_flag_ecn	Binary	ECN Echo TCP flag	✓		

Hindy, Bures, Tachtatzis & Bellekens, 2020, p. 4

MQTT IDS Dataset Details & Features, cont'd

Table 1 continued

Feature	Data Type	Description	Packet	Uni-flow	Bi-flow
tcp_flag_urg	Binary	Urgent TCP flag	✓		
tcp_flag_ack	Binary	Acknowledgement TCP flag	✓		
tcp_flag_push	Binary	Push TCP flag	✓		
tcp_flag_reset	Binary	Reset TCP flag	✓		
tcp_flag_syn	Binary	Synchronization TCP flag	✓		
tcp_flag_fin	Binary	Finish TCP flag	✓		
num_pkts	Integer	Number of Packets in the flow		✓	*
mean_iat	Decimal	Average inter arrival time		✓	*
std_iat	Decimal	Standard deviation of inter arrival time		✓	*
min_iat	Decimal	Minimum inter arrival time		✓	*
max_iat	Decimal	Maximum inter arrival time		✓	*
num_bytes	Integer	Number of bytes		✓	*
num_psh_flags	Integer	Number of push flag		✓	*
num_rst_flags	Integer	Number of reset flag		✓	*
num_urg_flags	Integer	Number of urgent flag		✓	*
mean_pkt_len	Decimal	Average packet length		✓	*

Hindy, Bures, Tachtatzis & Bellekens, 2020, p. 4

MQTT IDS Dataset Details & Features, cont'd

std_pkt_len	Decimal	Standard deviation packet length		✓	*
min_pkt_len	Decimal	Minimum packet length		✓	*
max_pkt_len	Decimal	Maximum packet length		✓	*
mqtt_messagetype	Integer	MQTT message type	✓		
mqtt_messagelength	Binary	MQTT message length	✓		
mqtt_flag_uname	Binary	User Name MQTT Flag	✓		
mqtt_flag_passwd	Binary	Password MQTT flag	✓		
mqtt_flag_retain	Binary	Will retain MQTT flag	✓		
mqtt_flag_qos	Integer	Will QoS MQTT flag	✓		
mqtt_flag_willflag	Binary	Will flag MQTT flag	✓		
mqtt_flag_clean	Binary	Clean MQTT flag	✓		
mqtt_flag_reserved	Binary	Reserved MQTT flag	✓		
is_attack	Binary	1 if the instance represents an attack, 0 otherwise.	x	x	x

Hindy, Bures, Tachtatzis & Bellekens, 2020, p. 4

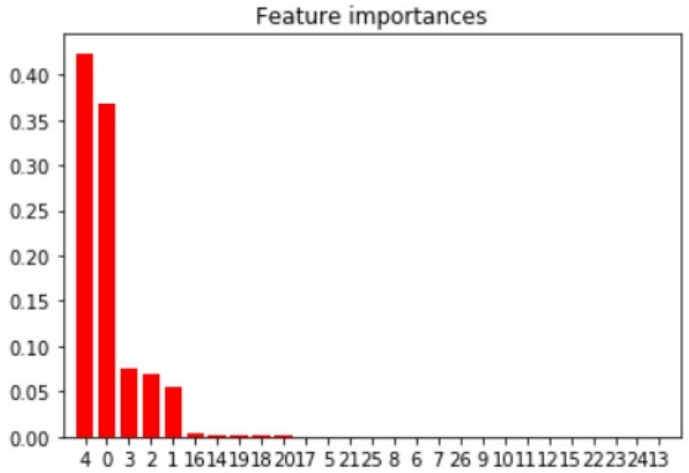
Feature Selection

Feature Selection

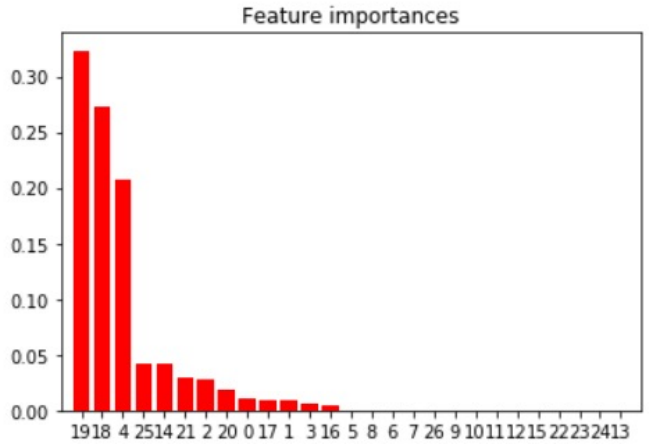
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import SelectFromModel
sel = SelectFromModel(RandomForestClassifier(n_estimators = 100))
```

Attack Type	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Brute Force	'src_ip’	'dst_ip',	'protocol'	'ttl'	'ip_len'
Sparta	'ip_len'	'tcp_flag_push	'mqtt_messagetype	'mqtt_messagelength'	'mqtt_flag_clean'
Aggressive Scan (Scan_A)	'ttl’	'ip_len’	'ip_flag_df'	-	-
UDP scan (Scan_sU)	'ttl’'	'ip_len'	'ip_flag_df'	-	-

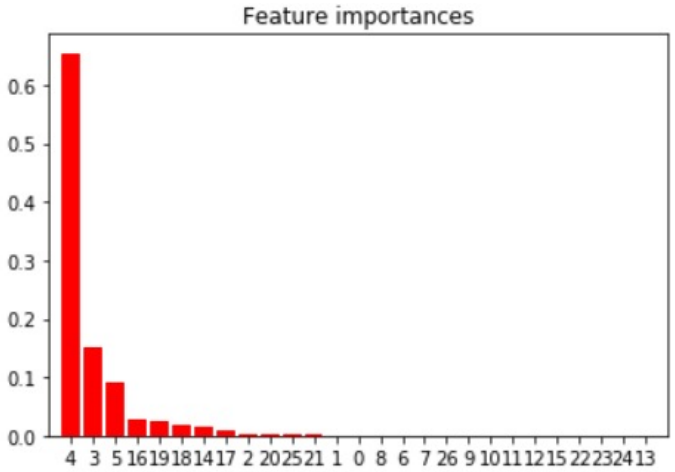
Feature Selection Graphs



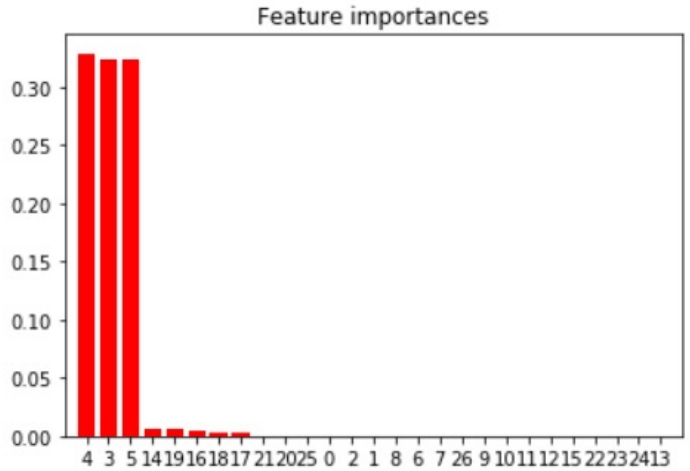
MQTT Brute Force



Sparta



Aggressive Scan_A



UDP Scan_sU

Methodology

Machine Learning Libraries & Algorithms

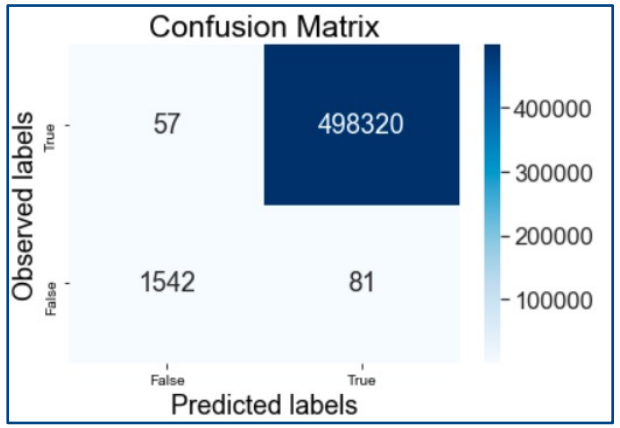
- Python 3.8, Scikit-Learn version 0.23.2
  - Random Forest
  - Logistic Regression
  - Decision Tree
  - K-Nearest Neighbors
  - Support Vector Machine
- StandardScaler
- Shuffle
- K-Fold Cross Validation (5-fold)

RESULTS

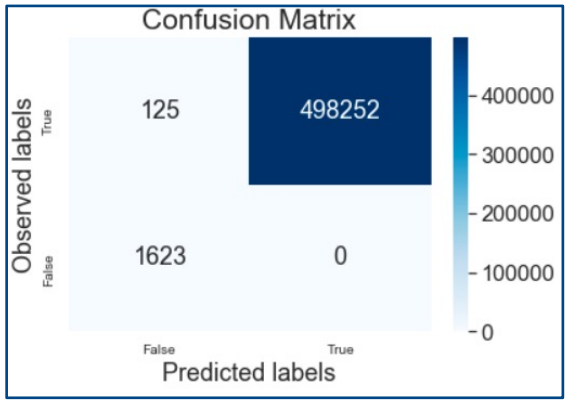
MQTT brute-force attack (MQTT BF)

	Accuracy	Precision	Recall	F1
Random Forest	99.97	99.98	99.99	99.98
Logistic Regression	99.35	99.94	99.98	99.96
Decision Tree	99.92	99.92	1	99.96
K-Nearest Neighbors	99.97	99.96	1	99.98
Support Vector Machine	99.97	99.99	99.97	99.98

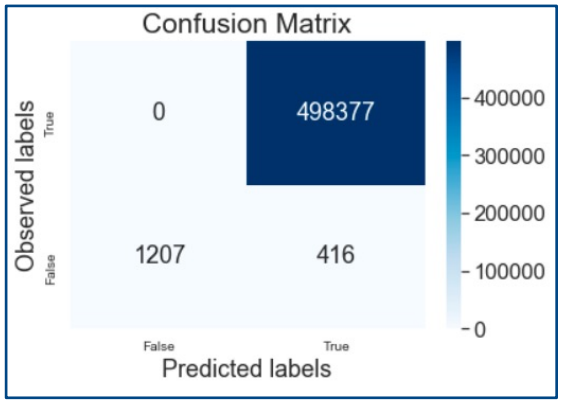
MQTT brute-force attack (MQTT BF) Confusion Matrices



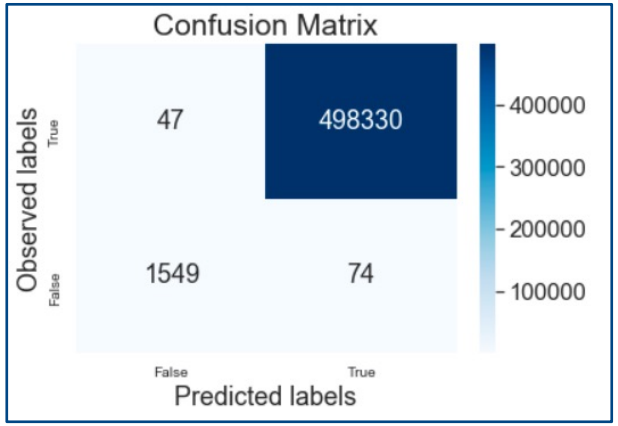
Random Forest



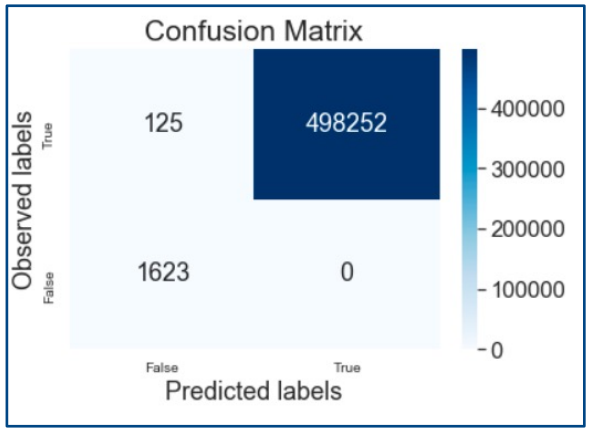
Logistic Regression



Decision Tree



KNN

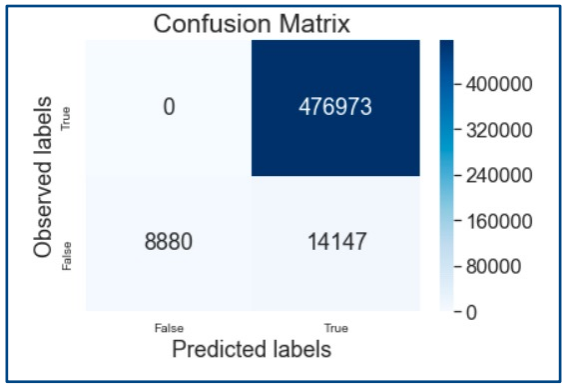


Support Vector Machine

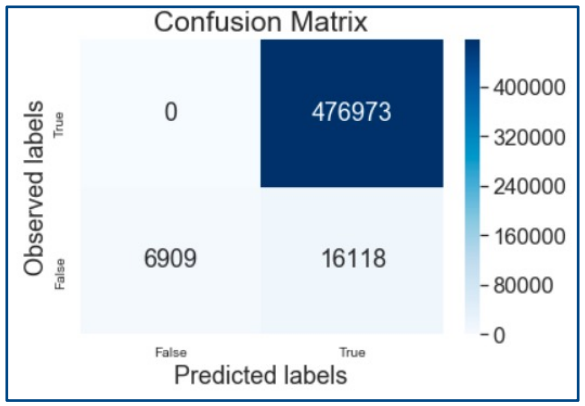
Sparta SSH brute-force (Sparta)

	Accuracy	Precision	Recall	F1
Random Forest	97.17	97.12	1.0	98.53
Logistic Regression	96.77	96.73	1.0	98.35
Decision Tree	97.12	97.05	1.0	98.517
K-Nearest Neighbors	97.17	97.12	1.0	98.53
Support Vector Machine	96.77	96.73	1.0	98.34

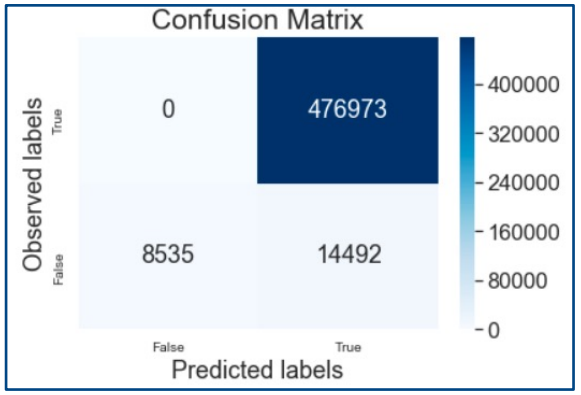
Sparta SSH brute-force (Sparta) Confusion Matrices



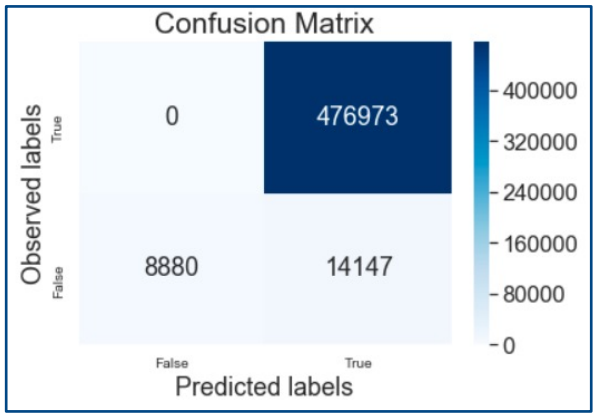
Random Forest



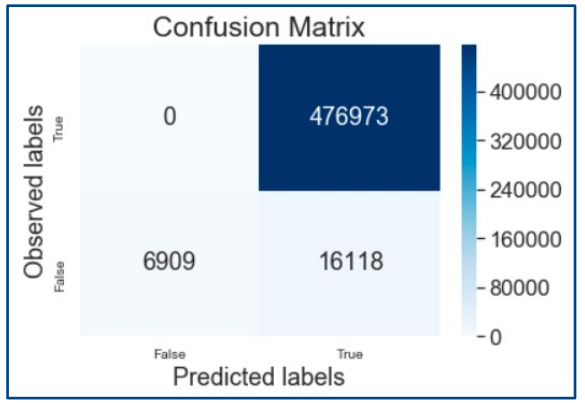
Logistic Regression



Decision Tree



KNN

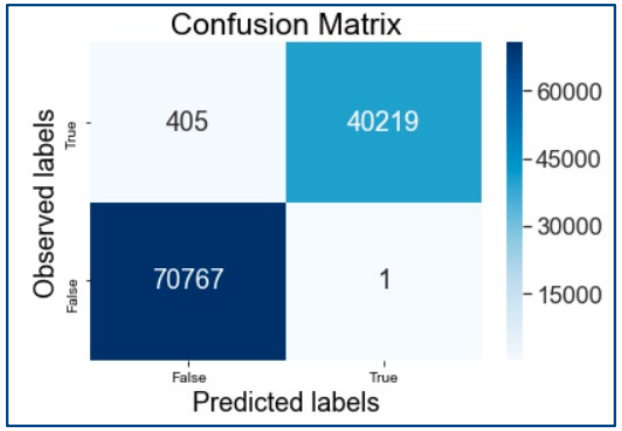


Support Vector Machine

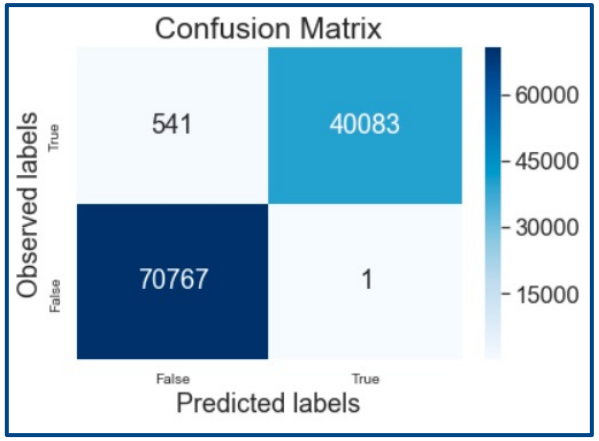
Aggressive scan (Scan A)

	Accuracy	Precision	Recall	F1
Random Forest	99.95	99.98	1	99.99
Logistic Regression	99.87	99.98	1	99.84
Decision Tree	99.95	99.98	1	99.81
K-Nearest Neighbors	99.95	99.98	1	99.99
Support Vector Machine	99.91	99.98	99.75	99.87

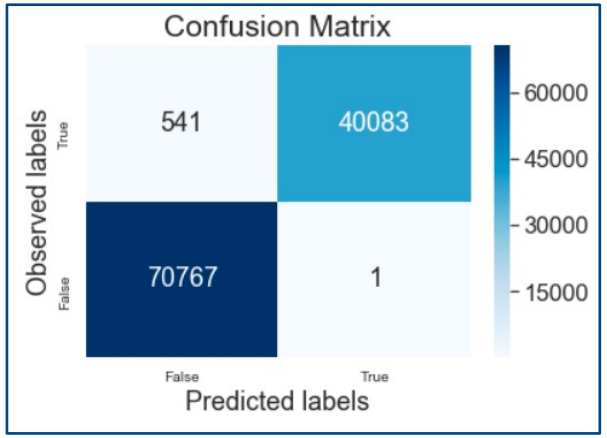
Aggressive scan (Scan A)



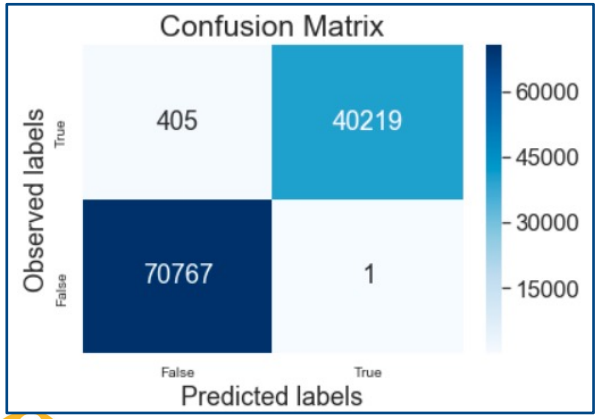
Random Forest



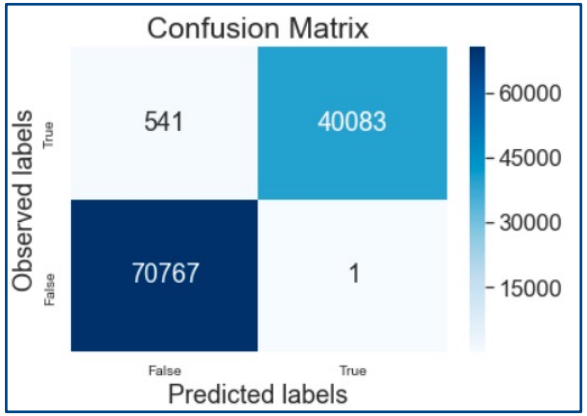
Logistic Regression



Decision Tree



KNN

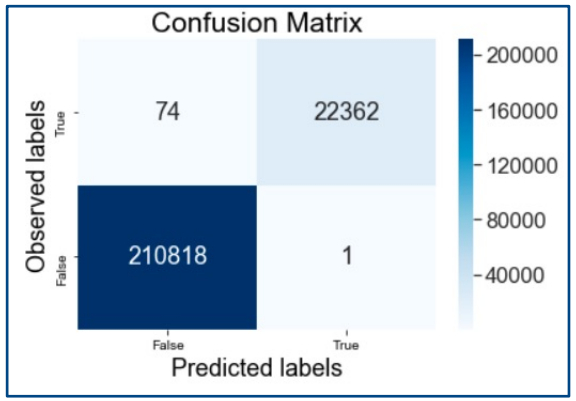


Support Vector Machine

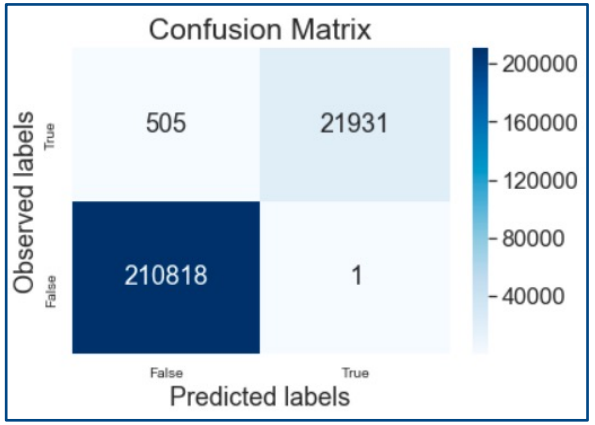
User Datagram Protocol (UDP) scan (Scan sU)

	Accuracy	Precision	Recall	F1
Random Forest	99.99	1.0	1.0	99.99
Logistic Regression	99.89	99.98	1.0	99.85
Decision Tree	99.98	1.0	1.0	99.93
K-Nearest Neighbors	99.95	1.0	1.0	99.99
Support Vector Machine	99.91	99.98	99.97	99.95

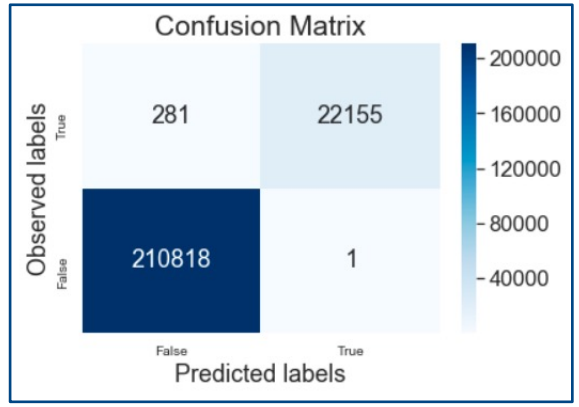
UDP scan (Scan sU) Confusion Matrices



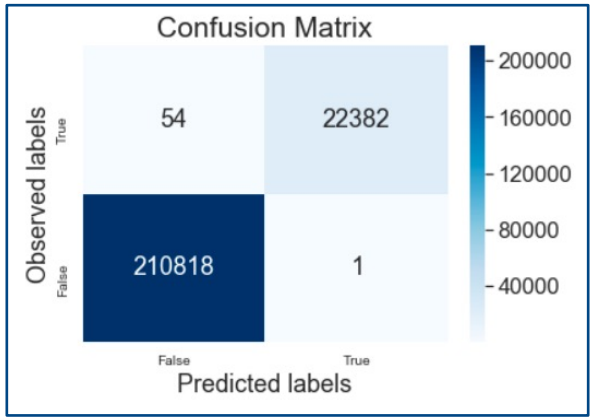
Random Forest



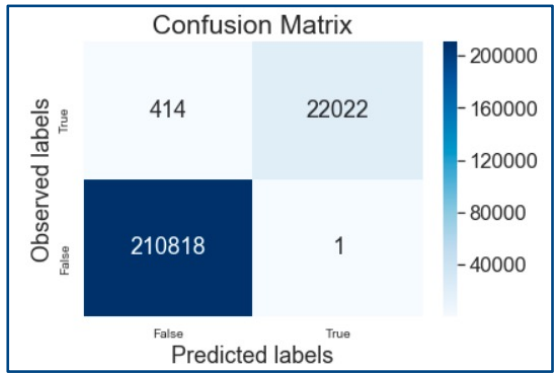
Logistic Regression



Decision Tree



KNN



Support Vector Machine

Future Work

Future Work

- Perform more missing value/substitution analysis on dataset
- Include recent Naïve Bayes classifier results (much lower accuracy) to this research
- Run DNN against 4 attack datasets
- Perform analysis of the BoT_IoT, IoT_23 and ToN_IoT datasets with this model

The IoT Network Intrusion Dataset

IoT Network Intrusion Dataset

- **Purpose:** This dataset provides researchers a means to evaluate normal versus malicious traffic in IoT (Internet of Things) network infrastructures via machine learning and deep learning algorithms.

- In recent years, machine learning algorithms have been found effective in combating cyberattacks, but much of that research has been conducted against traditional network environments.

- As the use IoT devices has exploded, and will continue to grow at an exponential rate, evaluating normal IoT network traffic and devices behaviors is necessary to determine how effective ML and DL algorithms will be in detecting (classifying) malicious network activity.

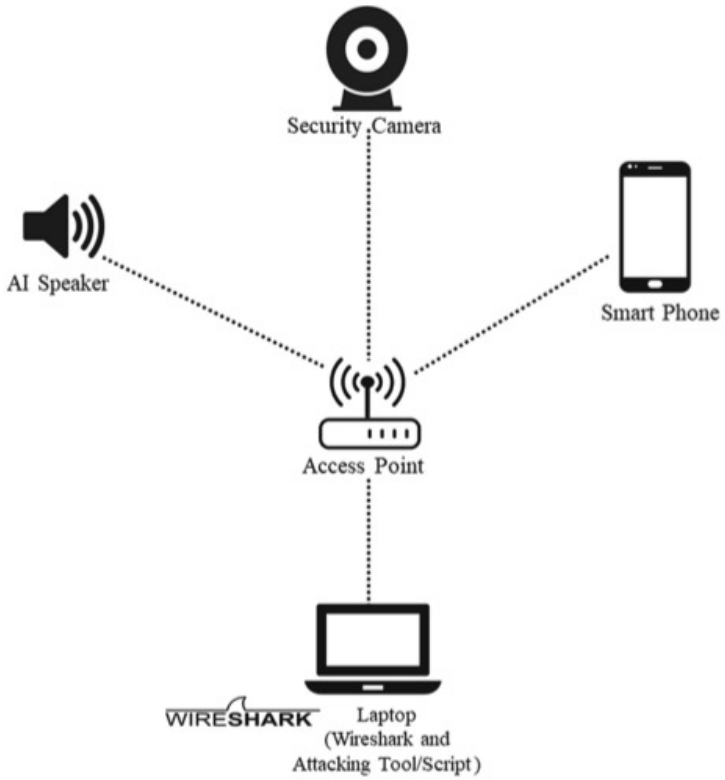
- IoT devices are the most vulnerable nodes in modern networks, as they exist at the edge of networks and have constrained architectures.

- The dataset was made publicly available on 09/2019 and can be retrieved from the IEEE dataport page: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset

IoT Network Environment & Collection

- **Devices:**
  - Smart home device SKT NGU (IoT victim device)
  - EZVIZ Wi-Fi camera (IoT victim device)
  - Other devices connected to the smart home router include laptops, tablets, smartphones – attacking devices

- **Data Collection:**
  - CICflowmeter – used to capture pcap files from the network and .csv files were generated as a result



Goutte & Zhu, 2019, p. 512

IoT Network Intrusion Dataset Details

- **# of Columns:** 86 network features
- **# of Rows:** 625783 rows of data
- **Labels/Target Columns:**
  - **"Label"** – Normal, Anomaly
  - **"Cat"** – Normal, DoS, Mirai, MITM (man in the middle), Scan
  - **"Sub Cat"** – Normal, Syn Flooding, Brute Force, HTTP Flooding, UDP Flooding, ARP Spoofing, Host Port, OS

Data Preparation & Preprocessing

Data Prep. & Preprocessing

- **Removed Highly Correlated Features** (as suggested by the dataset originator): *Active_Max, Bwd_IAT_Max, Bwd_Seg_Size_Avg, Fwd_IAT_Max, Fwd_Seg_Size_Avg, Idle_Max, PSH_Flag_Cnt, Pkt_Size_Avg, Subflow_Bwd_Byts, Subflow_Bwd_Pkts, Subflow_Fwd_Byts, Subflow_Fwd_Pkts*
- There were no NaN (missing) values but there were several INF (infinite) values that were removed
- The features were normalized

```
# print out columns with infinite values and convert existing ones to NaN values
dataset.columns.to_series()[np.isinf(dataset).any()]
dataset = dataset.replace([np.inf, -np.inf], np.nan)

# re-Check NaN values and then drop them
dataset.isnull().values.any()
np.isnan(dataset).any()
dataset = dataset.dropna() # Drop NaN values
dataset
```

```
# Normalize the data
from sklearn.preprocessing import MinMaxScaler
scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)
X_test = scaling.transform(X_test)
```

Data Modeling & Results

Random Forest Results

Accuracy: 90.57%

```
In [150]: rnd_clf.score(X_test, y_test)
Out[150]: 0.9056066771663616
```

- The Random Forest Classifier Algorithm was run against the "Cat" target column which had 5 categories

Modeling

```
# Data Modeling
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

# Normalize the data
from sklearn.preprocessing import MinMaxScaler
scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)
X_test = scaling.transform(X_test)

# Run Random Forest; create an object then
# Checking for data type issues: https://pandas.pydata.org/pandas-docs/stable/
reference/api/pandas.DataFrame.astype.html
# Convery Pandas dataframe to Numpy: https://stackoverflow.com/questions/13187778/
convert-pandas-dataframe-to-numpy-array

rnd_clf = RandomForestClassifier(n_estimators=500, max_leaf_nodes=16, n_jobs=-1)
rnd_clf.fit(X_train, y_train.values.ravel())
```

Support Vector Machine Results

Accuracy: 96.76%

```
In [61]: svm_cls.score(X_test, y_test)
Out[61]: 0.9675655364837747
```

- The Support Vector Machine Classifier Algorithm was run against the "Label" target column which had 2 categories (binary)

Modeling

```
# Data Modeling
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)

# Nomralize the data
from sklearn.preprocessing import MinMaxScaler
scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
X_train = scaling.transform(X_train)
X_test = scaling.transform(X_test)

# Run Random Forest; create an object then
# Checking for data type issues: https://pandas.pydata.org/pandas-docs/stable/
reference/api/pandas.DataFrame.astype.html
# Convery Pandas dataframe to Numpy: https://stackoverflow.com/questions/13187778/
convert-pandas-dataframe-to-numpy-array


# SMV Multi-variate Classifier
from sklearn import model_selection
from sklearn.svm import SVC
svm_cls = SVC(kernel='linear', cache_size=7000)
svm_cls.fit(X_train, y_train)
svm_cls.score(X_test, y_test)
```

Deep Learning Neural Network

Modeling

- The Deep Learning Neural Network Algorithm was run against the “Cat” target column which had 5 categories

```
import keras
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
n_cols = X.shape[1]
n_cols

model.add(Dense(5, activation='relu', input_shape=(n_cols,)))
model.add(Dense(5, activation='relu'))
model.add(Dense(5, activation='softmax'))

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10)

pred_train= model.predict(X_train)
scores = model.evaluate(X_train, y_train, verbose=0)
print('Accuracy on training data: {}% \n Error on training data:
{}'.format(scores[1], 1 - scores[1]))

pred_test= model.predict(X_test)
scores2 = model.evaluate(X_test, y_test, verbose=0)
print('Accuracy on test data: {}% \n Error on test data: {}'.format(scores2[1], 1 -
scores2[1]))
```

Deep Learning Neural Network Results

Accuracy: 81.2%

```
In [125]: model.fit(X_train, y_train, epochs=10)
Epoch 1/10
500332/500332 [==============================] - 16s 32us/step - loss: 0.6244 - accuracy: 0.7446
Epoch 2/10
500332/500332 [==============================] - 15s 30us/step - loss: 0.5321 - accuracy: 0.7807
Epoch 3/10
500332/500332 [==============================] - 16s 31us/step - loss: 0.5132 - accuracy: 0.8050
Epoch 4/10
500332/500332 [==============================] - 15s 31us/step - loss: 0.5085 - accuracy: 0.8076
Epoch 5/10
500332/500332 [==============================] - 16s 31us/step - loss: 0.5066 - accuracy: 0.8085
Epoch 6/10
500332/500332 [==============================] - 15s 30us/step - loss: 0.5051 - accuracy: 0.8084
Epoch 7/10
500332/500332 [==============================] - 15s 31us/step - loss: 0.5037 - accuracy: 0.8087
Epoch 8/10
500332/500332 [==============================] - 15s 31us/step - loss: 0.5028 - accuracy: 0.8086
Epoch 9/10
500332/500332 [==============================] - 15s 30us/step - loss: 0.5014 - accuracy: 0.8092
Epoch 10/10
500332/500332 [==============================] - 15s 31us/step - loss: 0.5004 - accuracy: 0.8093
Out[125]: <keras.callbacks.callbacks.History at 0x1d102ad0a48>
```

```
Accuracy on training data: 0.8119408488273621%
 Error on training data: 0.18805915117263794
```

```
Accuracy on test data: 0.8122047185897827%
 Error on test data: 0.18779528141021729
```

Future Work

Future Work

- Create data exploration graphics to better understand the dataset
- Run Random Forest and DNN against the Sub_Cat target column
- Create evaluation metrics/visualizations for Precision, Recall, F-score, Confusion Matrix, ROC and AUC

Acknowledgments

- NSA
- Dr. Kaushik Roy, Associate Professor, Department of Computer Science, Director, Center for Cyber Defense (CCD), North Carolina Agricultural and Technical State University
- Mr. Zhipeng Liu, PhD Candidate, Center for Cyber Defense (CCD), North Carolina. Agricultural and Technical State University
- Dr. Niraj Thapa, Research Associate, Center for Cyber Defense (CCD), North Carolina. Agricultural and Technical State University

References

- Canadian Institute of Cybersecurity. (2019). DDoS Evaluation Dataset (CICDDoS2019). [Online]. New Brunswick University, Canadian Institute of Cybersecurity (CIC). Available https://www.unb.ca/cic/datasets/ddos-2019.html
- J. Cañedo and A. Skjellum, "Using machine learning to secure IoT systems," *2016 14th Annual Conference on Privacy, Security and Trust (PST)"*, Auckland, 2016, pp. 219-222.
- G. Falco, C. Caldera and H. Shrobe, "IIoT cybersecurity risk modeling for SCADA systems," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4486-4495, Dec. 2018. doi: 10.1109/JIOT.2018.2822842
- C. Goutte and X. Zhu (Eds.), *Canadian AI 2020, LNAI 12109*, pp. 508–520, 2020. https://doi.org/10.1007/978-3-030-47358-7_52
- M. S. Harsha, B. M. Bhavani and K. R. Kundhavai, "Analysis of vulnerabilities in MQTT security using shodan API and implementation of its countermeasures via authentication and ACLs," in - *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI),* 2018, . DOI: 10.1109/ICACCI.2018.8554472
- H. Hindy *et al*, "Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoT-IDS2020 Dataset)," unpublished.
- S. Hou and X. Huang, "Use of machine learning in detecting network security of edge computing system," *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, Suzhou, China, 2019, pp. 252-256. doi: 10.1109/ICBDA.2019.8713237
- F. Hussain *et al*, "Towards a universal features set for IoT botnet attacks detection," in - *2020 IEEE 23rd International Multitopic Conference (INMIC),* 2020, . DOI: 10.1109/INMIC50486.2020.9318106.
- N. Koroniotis, N. Moustafa, E. Sitnikova, E., and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems", *University of New South Wales Canberra,* Australia, 100, pp. 779-796. doi:10.1016/j.future.2019.05.041. Available: https://arxiv.org/pdf/1811.00701.pdf
- Y. Lu and L. D. Xu, "Internet of Things (IoT) cybersecurity research: A review of current research topics," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2103-2115, April 2019. doi: 10.1109/JIOT.2018.2869847
- M. Mamdouh, M. A. I. Elrukhsi and A. Khattab, "Securing the Internet of Things and wireless sensor networks via machine learning: A survey," 2018 International Conference on Computer and Applications (ICCA), Beirut, 2018, pp. 215-218.

References

- F. Meneghello, M. Calore, D. Zucchetto, M. Polese and A. Zanella, "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT Devices," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8182-8201, Oct. 2019. doi: 10.1109/JIOT.2019.2935189
- L. Perry Lea, IoT and Edge Computing for Architects - Second Edition: Implementing IoT Solutions with Sensors, Communication, Edge Computing, Networking, Analytics and Security. 2020Available: http://www.vlebooks.com/vleweb/product/openreader?id=none&isbn=9781839218873.
- M. Roopak, G. Yun Tian and J. Chambers, "Deep learning models for cyber security in IoT networks," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 0452-0457. doi: 10.1109/CCWC.2019.8666588
- Yalin E. Sagduyu and Yi Shi and Tugba Erpek, "IoT network security from the perspective of adversarial deep learning," 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 2019, pp. 1-9. doi: 10.1109/SAHCN.2019.8824956
- I. Sharafaldin, A. Lashkari, S. Hakak, & A. Ghorbani, "Developing realistic distributed denial of Service (DDoS) attack dataset and taxonomy," New Brunswick University, Canadian Institute of Cybersecurity (CIC). Available https://ieeexplore.ieee.org/abstract/document/8888419
- A. Sivanathan et al., "Classifying IoT devices in smart environments using network traffic characteristics," IEEE Transactions on Mobile Computing, vol. 18, no. 8, pp. 1745-1759, 1 Aug. 2019.
- V. Tsiatsis, S. Karnouskos, J. Holler, D. Boyle and C. Mulligan. Internet of Things, Second Edition, London,UK: Academic Press, 2019.
- I. Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," IEEE Access, vol. 9, pp. 103906-103926, 2021. . DOI: 10.1109/ACCESS.2021.3094024.
- R. Vishwakarma and A. K. Jain, "A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 1019-1024
- L. Xiao, X. Wan, X. Lu, Y. Zhang and D. Wu, "IoT security techniques based on machine learning: How do IoT devices use AI to enhance security?" IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 41-49, Sept. 2018. doi: 10.1109/MSP.2018.2825478