MACHINE LEARNING CLASSIFICATION USING MOTIF BASED GRAPH DATABASES CREATED FROM UWF-ZEEKDATA22

Sikha S. Bagui [1], Dustin Mink [2], Subhash C. Bagui [3], Jadarius Hill [1], Farooq Mahmud [1] and Michael Plain [1]

[1] Department of Computer Science, University of West Florida, Pensacola, Florida, USA
[2] Department of Cybersecurity, University of West Florida, Pensacola, Florida, USA
[3] Department of Mathematics and Statistics, University of West Florida, Pensacola, Florida, USA

ABSTRACT

This study uses motif-based graph databases to visualize and classify tactics in the MITRE ATT&CK framework. Machine Learning classification models, capable of detecting Reconnaissance network attack tactics, labeled as per the MITRE ATT&CK framework, are created for the newly created UWF-ZeekData22 dataset. The work analyzes Zeek Connection logs. Feature selection is performed using graph motifs. Results show that model performance can be increased using various network graph motifs. Upon completion of this work, it was concluded that, of the motifs used, the Star motif performed the best; and, the most important feature for predicting Reconnaissance network attacks within the Zeek Connection Logs dataset was the "From" feature, or Source IP, which represents the network address from where the connection is originating. It was also determined that, irrespective of which motif was used to train the model, the Decision Tree algorithm performed best.

KEYWORDS

Graph Databases, Motifs, Star Motif, Reconnaissance, Machine Learning, Cybersecurity, Visualizing attacks

1. INTRODUCTION

Cybersecurity has become a critical issue within our society. With the increasing reliance on technology, the number of occurrences and complexities of cyberattacks increases. Cyberattacks can cause severe consequences such as legal liabilities, financial losses, and even the loss of lives. Cybersecurity Ventures estimates that by 2025, cybercrime costs will grow to approximately $10.5 trillion USD [1], which displays the urgent need for effective cybersecurity measures.

Cyberattacks can present themselves in various forms, such as ransomware, social engineering, hacking, phishing, and malware. These effects can target single individuals, businesses, and even whole governments alike, leaving devastating, long-lasting consequences. As the world continues to become digitized, it is imperative that we have the ability to protect ourselves against cyber

David C. Wyld et al. (Eds): CSITAI, MATHCS, ITCAU, COMSAP, ACINT - 2024
pp. 135-154, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.142510

threats; researchers have started to use graph databases coupled with machine learning (ML) models to create models capable of detecting nefarious network traffic.

Graph databases are a type of NoSQL database that store data in the form of vertices and edges. Vertices represent entities and the edges represent the relationships between them [2]. Graph databases allow for more flexible and efficient querying of more complex data relationships since graph databases do not use the traditional table and row architecture to store data [3]. Because of the looser data schemas, graph databases are naturally more suited for applications that have highly interconnected data, such as social networks, fraud detection systems, and recommendation engines; they also allow for more accurately representing network topologies as each network device can be represented as a vertex, with the edges between them acting as network routes.

Feature selection is one of the most important steps in ML model creation. Feature selection is when a subset of the dataset's features are selected to train the model. Feature selection can have a significant impact on model accuracy and can reduce the time needed to train the model, hence the time needed to detect attacks [4]. Using too many features at the same time can lead to incorrect predictions due to the curse of dimensionality [5]. There are many different methods of selecting features including Forward Feature Selection, Backward Feature Elimination, and Exhaustive Feature Selection [6].

The novelty of this work lies in using graph databases to visualize and classify the network attacks using different graph motifs. Since network data is interconnected, using graph databases to analyze and visualize the data was a natural selection. Feature selection was done using various graph motifs. In order to fully understand how the attacks are happening, where they are originating from, and where they are being targeted, it is important to visually analyze the attacks. In this work, a new network log dataset, UWF-Zeekdata22, composed of Zeek Connection (Conn) logs, defined as per the MITRE Adversarial Tactics, Techniques, and Common knowledge (ATT&CK) Framework [7], is used to classify and visualize the attack tactics in this data. Specifically, the work focuses on detecting the Reconnaissance tactic (TA0043) [8]. Various ML algorithms were used in an attempt to create the best ML model using the smallest number of features.

The rest of this paper is organized as follows. Section two presents the related works; section three presents the background necessary to understand this work, that is, the motifs and the machine learning algorithms; section four presents the data and pre-processing; section five presents the results and discussion and the last section presents the conclusions.

2. RELATED WORKS

The combination of graph databases and ML techniques has been a topic of interest for many researchers in various fields. In the financial industry, the two most commonly are paired to create ML models capable of detecting financial fraud such as credit card fraud [9,10]. In the hospitality industry, researchers have used ML models coupled with Neo4j, an open-source no-SQL graph database, to create a ML classification models that can predict hotel reservation cancellations [11]. Another set of researchers paired ML and graph databases to analyze social networks; these researchers used graph-based representations of social networks and ML algorithms to create a model capable of predicting the importance of nodes within the social network [12]. [13] looked at similarity graphs to determine how social networks differ across platforms and how information networks change over time. Other researchers extended graph databases and ML into the medical field [14]. This set of researchers created a model to predict autism and its conversion to Alzheimer's disease [15].

The idea of using graph databases with machine learners has started to become an item of interest for network security researchers. One group of researchers created attack profiles that were used to detect simulated network attacks. To do this, the researchers utilized a graph database, which allowed for the identification of the network attacker and the impacted components of the network [16]. [17] proposed ADAMM to look at finding anomalous instances using complex graph databases of node and edge attributed multi-graphs. They also used the associated metadata. [18] looked at finding anomalous activity given a set of node-labeled directed weighted graphs. They proposed GAWD for detecting anomalous graphs in directed weighted graph databases. [19] used Memgraph to apply graph ML techniques to detect cybersecurity attack tactics. Node classification was used to predict the connection between IP addresses and ports using Memgraph's graph neural networks.

The work detailed in this paper expands on the idea of extending graph databases, generated in [20], into the network security discipline. [20] visually represents the Reconnaissance Tactic of the MITRE ATT&CK framework using UWF-Zeekdata22, but does not go as far as making ML predictions.

This work, however, differs from previous research in various aspects. First and foremost, this work focuses on predicting one specific network attack tactic, Reconnaissance, using ML classification. Additionally, this work incorporates using graph network motifs within the ML models in an attempt to improve model prediction.

3. BACKGROUND

In this section we discuss the graph motifs as well as the ML classifiers to be used in this work.

3.1. Motifs

Graph motifs are recurring subgraph patterns between vertices found in complex graph networks [21]. Motifs are utilized to help understand the function and structure of complex networks. By understanding these patterns, one can learn valuable information about the dynamics of complex systems and can use that information to curate more effective strategies for network control and network analysis. In this research, two different motifs will be explored: star motifs and clique motifs.

3.1.1. Star Motif

Star motifs, by definition, are a pattern in which there is a central vertex that is connected to other outer vertices, resembling the shape of a star. Figure 1 presents an example of a generic star motif. There is a central vertex (143.88.2.10) that is connected to outer vertices (183.88.7.10, 183.88.7.15, 183.88.7.12, 183.88.7.11, and 183.88.7.1). This means that, the central vertex, 143.88.2.10, is generating or sending out attacks to the other vertices.

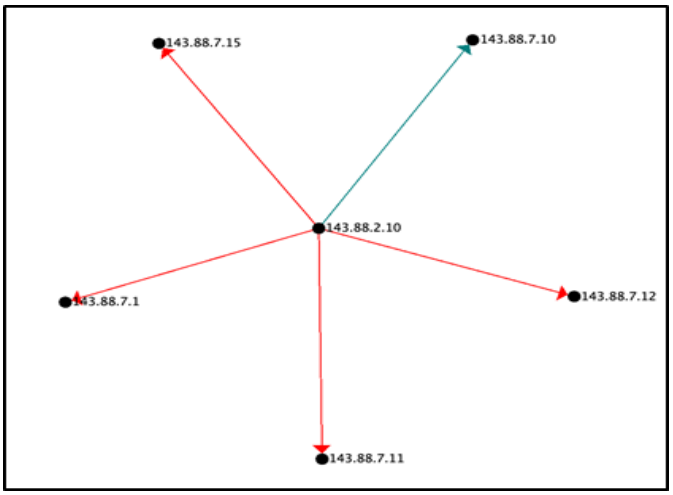

Figure 1. Example of a generic Star Motif

Figure 2 presents the star motif of the Reconnaissance attack tactic, found when analyzing the Reconnaissance data - the vertex with the IP address 143.88.2.10 acts as the central vertex with all connections originating from that central vertex. This means that all the attacks or reconnaissance is been done from (or originating from) the central vertex 143.88.2.10 and going to so many other vertices.

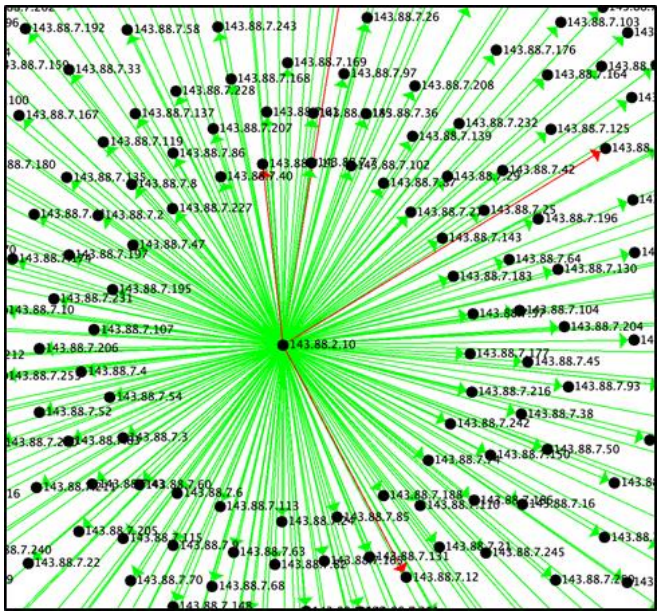

Figure 2. Star Motif for Reconnaissance

3.1.2. Clique Motif

Clique motifs are complete subgraphs, in which the subgraph has all adjacent vertices [22]. The subgraphs in cliques are relatively homogenous and have homogenous connectivity. Figure 3 presents an example of a clique motif. Each vertex in the graph is connected by an edge to every other vertex.

Figure 4 presents two examples of clique motifs found within the attack data. The yellow-highlighted clique consists of vertices 143.88.11.1, 143.88.11.13, and 143.88.11.10. The blue-highlighted clique consists of vertices 143.88.11.1, 143.88.11.12, and 143.88.11.10.

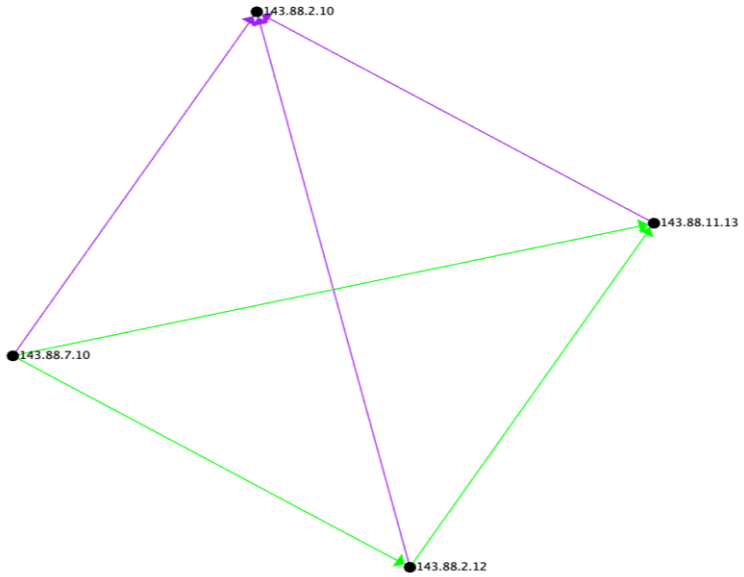

Figure 3. Clique Motif

3.2. Machine Learning Algorithms

The five algorithms that were implemented and evaluated for predicting Reconnaissance network attack tactics were Naive Bayes (NB), Decision Tree (DT), Support Vector Machines (SVM), Logistic Regression (LR), and Gradient Boosting Trees (GBT).

3.2.1. Naive Bayes

Naive Bayes is a supervised learning algorithm used for classification. The “naive” in the algorithm name comes from the fact that the algorithm assumes the input features are conditionally independent of each other (in reality features are rarely independent). This naive assumption makes calculating the conditional probability easier. To make a prediction, the algorithm computes the conditional probability of each class given the in-put features using Bayes' theorem. The class with the highest conditional probability is assigned the predicted class label [23].

3.2.2. Decision Tree

Decision trees are a supervised learning algorithm used for both classification and regression and resembles a tree-like structure. The objective of the Decision Tree algorithm is to maximize information gain at each split in the tree. Each branch represents an outcome of a test based on a feature of the training set. At each root vertex, the feature with the highest predictive power is used to split the tree. For each decision/internal vertex, a feature of the dataset is tested against a certain value and the tree is traversed based on the results. Once a leaf vertex is reached, a condition to stop splitting is achieved and a prediction is made [24].

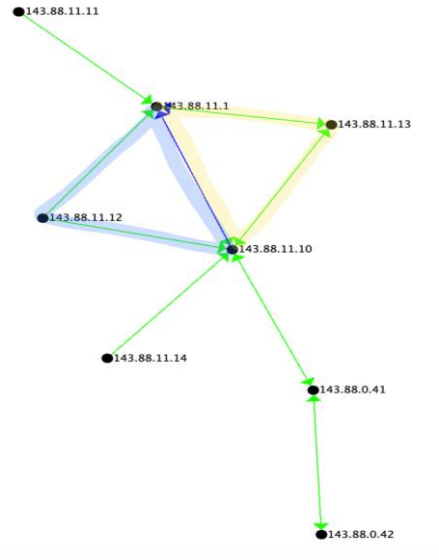

Figure 4. Clique Motif within the Data

3.2.3. Support Vector Machines

Support vector machines are a supervised learning algorithm that can be used for both classification and regression. SVM works by finding a linear decision boundary that maximizes the margin between classes - decision boundaries with large margins tend to have lower generalization error. If the data is not linearly separable, the data is converted to linear separable data in a higher dimensional space [25].

3.2.4. Logistic Regression

Logistic Regression, a supervised learning algorithm used for binary classification, models the probability of a binary response variable based on one or more predictor variables. The objective of the Logistic Regression algorithm is to fit a line that best separates the two classes in the data. It uses a logistic function to model the relationship between the predictor variables and the response variable. The logistic function takes in the linear combination of the predictor variables and returns the probability of the response variable being in a particular class [26].

3.2.5. Gradient Boosting Trees

Gradient Boosting Trees (GBT) are a supervised learning algorithm used for regression and classification. GBTs iteratively improve model performance by minimizing a loss function. A new weak learner is added to the model in each iteration to correct the errors made by previous learners. The final model is the sum of all weak learners and can be used to make predictions on new data [27].

4. THE DATASET: UWF-ZEEKDATA22

The dataset used in this work, UWF-ZeekData22, available at [28], is labeled as per the MITRE ATT&CK framework. It contains a total of 18,562,468 records with 9,280,869 of them representing network attacks tactics and 9,281,599 representing regular network traffic (benign traffic). UWF-ZeekData22 contains 10 tactics, presented in Table 1, with the highest number of

tactics being the Reconnaissance tactic [29]. Due to the minimal amount of data for attack tactic types other than Reconnaissance, this research is focused only on the Reconnaissance tactic.

Table 1. Tactics in UWF-ZeekData22 [29]

Attack Tactic	**Count**
None (Not an attack)	9,281,599
Reconnaissance	9,278,722
Discovery	2,086
Credential Access	31
Privilege Escalation	13
Exfiltration	7
Lateral Movement	4
Resource Development	3
Defense Evasion	1
Initial Access	1
Persistence	1

The Zeek Connection logs dataset, crafted by the Cyberrange at the University of West Florida, has the schema presented in Table 2. The datasets used in this research are a sub-set of this Zeek Connection Logs dataset, also used in [20].

Table 2. Zeek Connection Logs Data Schema [30]

Attribute Name	**Attribute Description**
ts	Time of first packet
uid	Unique identifier of connection
id.orig_h	IP address of packet senders
id.orig_p	Outgoing port number
id.resp_h	IP address of packet receiver
id.resp_p	Incoming port number
proto	Transport layer protocol of connection
service	Identification of application protocol being sent over connection
duration	How long connection lasted
orig_bytes	Number of payload bytes originator sent
resp_bytes	Number of payload bytes responder sent
conn_state	Possible connection states
local_orig	If connection is responded to locally, value is T
local_resp	If connection is responded to locally, this value is T
missed_bytes	Number of bytes missed in content gaps, representative of packet loss
history	Records the state history of connections as a string of letters
orig_pkts	Number of packets originator sent. Set if: zeek:id:use_conn_size_analyzer = T
orig_ip_bytes	Number of packets responder sent. Set if: zeek:id:use_conn_size_analyzer = T
resp_pkts	Number of IP level bytes responder sent. Set if zeek:id:use_conn_size_analyzer = T
resp_ip_bytes	Number of IP level bytes responder sent
community_id id	Connection's 4-tuple of endpoint addresses/ports
tunnel_parents	If connection was over tunnel, indicates *uid* values for encapsulating parent(s) connections used over lifetime of inner connection

4.1. Data Preprocessing

Since the idea on the graph databases and in the graph motifs is to see where the attacks are originating from and where they are going, the attributes used in the graph formation, as per [20], were the source and destination IP addresses, bytes transferred, and connection duration. Table 3 shows these attributes with some sample data. Source and destination IP addresses were used as a key and the bytes transferred and connection duration were aggregated. Additional attributes were derived from the aggregated attributes: Count, Average Duration and Average bytes.

- Count is the number of connection occurrences between the source and destination IPs;
- Average duration is the average duration of all connections between the source and destination IPs;
- Average bytes is the bytes transferred for all connections between the source and destination IPs.

These attributes are presented in Table 4, with some sample data.

For data aggregation, the following pseudocode (Code snippet 1) was implemented in Java using linked hash maps; the key to the hash maps were the source and destination IP addresses. The values to the respective keys were the various attributes. Tables 3 and 4 show an example data input and data aggregation respectively.

```
For each record in the input file
       key = srcIP,dstIP
       if key has not been encountered before
          set key:connectionCount to 1
          set key:bytes       to record:bytes
          set key:duration    to record:duration
       else
          increment key:connectionCount
          add record:bytes    to key:bytes
          add record:duration to key:duration
```

Code Snippet 1.

Table 3. Example input data

Source IP	Destination IP	Bytes	Duration	Tactic
143.88.2.10	143.88.7.15	0	143.88.7.15	Reconnaissance
143.88.2.10	143.88.7.15	0	143.88.7.15	Reconnaissance
143.88.2.10	143.88.7.15	0	143.88.7.15	Reconnaissance
143.88.2.10	143.88.7.1	0	143.88.7.1	Reconnaissance

Table 4. Resulting aggregation

Source IP	Destination IP	Count	Bytes	Duration	Avg Bytes	Avg Duration	Tactic
143.88.2.10	143.88.7.15	3	0	0.02144094	0	108	362
143.88.2.10	143.88.7.1	1	0	0.00019192	0	15	18

4.2. Graph Datasets Generated

Table 5 provides an overview of the three graph datasets used to make predictions. Preprocessed data matching the star and clique motifs were extracted. In addition, a third dataset was created, Non-attack-edges. This dataset was a complete set of all the Reconnaissance and Non-Attack connections that occurred irrespective of the motifs.

The clique data had only 18 rows. This would not be sufficient to get meaningful ML results, hence this dataset was excluded from the ML analysis. Subsequent sections will discuss the results of running the algorithms against the star-edges and the non-attack-edges datasets.

Table 5. Graph Datasets Used

Dataset	**Description**	**Reconnaissance Count**	**Non-Attack Count**	**Raw Count**
Star-edges	The source and destination IP address represent edges that are in star motif	254	108	362
Clique-edges	The source and destination IP address represent edges that are in clique motif	3	15	18
Non-attack-edges	The source and destination IP address represent edges in a graph	258	118	376

4.3. Graph Dataset Schema

Table 6 shows the final graph dataset schema that was used in this ML analysis. The third column shows whether this was an original attribute or derived. Since, in a graph database analysis, it is important to visualize where the attacks are originating from and who they are targeting, hence the source and destination IPs were the main attributes. The data were grouped by source and destination IP address. The count was used to determine if there was an edge between the source and destination vertex pair. The other pieces of information that would be important would be, average and total duration between the source and destination vertex pairs and the average and total bytes between the source and destination vertex pairs. Another attribute related to this information was the hop count, that is, the distance from the root vertex to the destination vertex. So basically, this ML study would determine if, a dataset composed of where was the attack originating from, where was it going, how long was it taking, and how many bytes was it carrying, would be adequate to predict an attack.

Table 6. Graph Dataset Schema Used in Machine Learning Analysis

Column	Description	Source or Derived
Id	The edge's unique id. Not used in ML algorithm features	-
From	The edge's source vertex IP address	Source - id.orig_h
To	The edge's destination vertex IP address	Source - id.resp_h
Count	Count of the occurrences of an edge between the source and destination vertex pair	Derived
Avg_Duration	Average duration of all edge occurrences between source and destination vertex pair	Derived
Total_Duration	Total duration of the all edge occurrences between the source and destination pair	Derived
Avg_Bytes	Average bytes transferred in all edge occurrences between the source and destination vertex pair	Derived
Total_Bytes	Total sum of bytes transferred in all edge occurrences between the source and destination vertex pair	Derived
Hop_Count	Vertex distance from root vertex to destination vertex, starting with root as 1	Derived
Tactic	Attack tactic is either "Reconnaissance" or "None"	Source - tactic

5. RESULTS AND DISCUSSION

The Naive Bayes, Decision Tree, Logistic Regression, SVM, and Gradient Boosting Tree algorithms were run using the graph datasets in order to predict the attack tactic. For each algorithm, the results are presented for the Star-edges dataset and the Non-attack edges datasets. The objective was to predict a reconnaissance attack using the Star-edges and the Non-attack edges (which had some non-Reconnaissance edges mixed in).

For all datasets, all 255 feature combinations were evaluated and the best results as well as worst results are presented. The worst results were presented to further understand or verify the important features.

99% of the tactics present in the dataset were Reconnaissance, thus only the Reconnaissance tactic was analyzed. Binary classification was used, hence either a Reconnaissance attack happened or no attack happened. Table 7 lists the algorithm evaluation metrics used in determining the algorithms' performance.

Table 7. Algorithm Evaluation Metrics Used

Metric	Description
Accuracy	The percentage of data whose attack (or lack thereof) was correctly predicted.
Precision	A higher precision score indicates the algorithm results in fewer false positives.
Recall	A higher recall score indicates the algorithm can identify most of the positive cases.
F Measure	A metric that combines precision and recall to evaluate the performance of a binary classification.
False Positive Rate	The percentage of false positive cases.
AUC-ROC	A higher AUC-ROC score indicates more true positives and less false positives.
Confusion matrix	A matrix showing the number of true positive, false positive, false negative, and true negative cases. The matrix is read top to bottom, left to right.

5.1. Naïve Bayes Results

5.1.1. Star-Edges Dataset

With the Star-edge dataset, we are predicting whether a reconnaissance will be happening using the Naïve Bayes algorithm.

Table 8 presents the features that resulted in the best statistical metrics for the Star-edges using Naïve Bayes classification and Table 9 presents the features that resulted in the worst statistical metrics for the Star-edges using the Naïve Bayes classification. Although there were a few combinations that presented good results, for example [From and Average Duration] and [From, Average_Duration and Hop_count], the combination of the three features, [From, Total_Duration and Total_Bytes], as well as the combination of these latter three features with any other features, resulted in the best results. The confusion matrix shows that there were no false positives or false negatives.

From Table 9 it can be observed that the [From] feature by itself did not give good results using the Naïve Bayes classifier using Star-edges, and the combination of the [From] feature that do not also include the [Total_Duration, and Total_Bytes] features resulted in the lowest accuracy and other statistical metrics.

In summary, this means that, Source ID [From] by itself is not a good predictor of an attack or reconnaissance happening, but Source in combation with Average Duration or Total Duration and Hop Count are excellent indicators of reconnaissance happening.

Table 8. Naive Bayes Results: Star-Edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From, Avg_Duration	1	1	1	1	0	1	[78, 0 0, 33]
From, Avg_Duration, Hop_Count	1	1	1	1	0	1	[78, 0 0, 33]
From, Total_Duration, Hop_Count	1	1	1	1	0	1	[78, 0 0, 33]
From, Total_Duration, Avg_Bytes	1	1	1	1	0	1	[78, 0 0, 33]
From, Total_Duration, Total_Bytes	1	1	1	1	0	1	[78, 0 0, 33]
From, Total_Duration, Total_Bytes + Any other feature(s)	1	1	1	1	0	1	[78, 0 0, 33]

Table 9. Naive Bayes Results: Star-Edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From	0.58	0	1	0.92	0	0.5	[78, 0 33, 0]
From + Any other features except Total_Duration, Total_Bytes	0.58	0	1	0.92	0	0.5	[78, 0 33, 0]

5.1.2. Non-Attack-Edges Set

With the Non-attack edges dataset, we are predicting if reconnaissance happens even if some non-attack edges are mixed in.

Table 10 presents the features that resulted in the best statistical metrics for the Non-attack-edges dataset using Naïve Bayes classification and Table 11 presents the features that resulted in the worst statistical metrics for the Non-attack-edges dataset using Naïve Bayes classification for the Reconnaissance Tactic. The combination of the [From and Count] features resulted in the best accuracy with the fewest number of features. Adding additional features to [From and Count] kept the accuracy and other matrices the same. As Table 11 shows, utilizing only the [Hop_count] feature resulted in the lowest accuracy. So, the reconnaissance can also be predicted with some non-attack edges mixed in, with the Source IP and the Count features, and the Hop_count is the worst at predicting reconnaissance.

Table 10. Naive Bayes: Non-attack-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From, Count	0.983	0.944	0.976	0.980	0.024	0.988	[80, 2 0, 34]
From, Count + Any other feature(s)	0.983	0.944	0.976	0.980	0.024	0.988	[80, 2 0, 34]

Table 11. Naive Bayes: Non-attack-edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Hop_count	0.541	0	1	0.911	0	0.5	[78, 0 38, 0]

5.2. Decision Tree Results

5.2.1. Star-Edges Dataset

Table 12 presents the features that resulted in the best statistical metrics for the Star-edges dataset using DT classification and Table 13 presents the features that resulted in the worst statistical metrics for the Star-edges dataset using DT classification for the Reconnaissance Tactic. From Table 12, it can be noted that the [From] feature was the only feature needed for 100% accuracy by the DT algorithm. Adding additional features did not increase or decrease the accuracy. The confusion matrix shows that there were no false positives or false negatives. The difference in the magnitude of the DT and NB confusion matrices is due to the differences in training and validation set sizes. There was a 83%/17% split with the DT compared to an 70%/30% split with the NB, SVM, and LR algorithms. The training set had to be increased in order for the DT to work due to the relative low number of rows in the dataset. Table 13 shows that, the [To] feature (and combinations including the [To] feature) including the [To] feature without the [From] feature resulted in the lowest accuracy.

In summary, using the Star-edges, the Source_IP is the best feature at determining the reconnaissance attack using the DT algorithm and the Destination_IP is the worst feature for determining the reconnaissance attack using the DT algorithm.

Table 12. Decision Tree Results: Star-Edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From	1	1	1	1	0	1	[46, 0 0, 19]
From + any other feature(s)	1	1	1	1	0	1	[46, 0 0, 19]

Table 13. Decision Tree Results: Reconnaissance: Star-Edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
To	0.07	0.2	0	0	1	0.5	[0, 52 0, 13]
To + any other feature(s)	0.07	0.2	0	0	1	0.5	[0, 52 0, 13]

5.2.2. Non-Attack-Edges Dataset

Table 14 presents the features that resulted in the best statistical metrics for the Non-attack-edges dataset using DT classification and Table 15 shows the features that resulted in the worst statistical metrics for the Non-attack-edges dataset using DT classification for the Reconnaissance Tactic. As can be seen from Table 14, the combination of [Avg_duration and Hop_count] resulted in the best accuracy with the fewest number of features. Adding additional features to [Avg_duration and Hop_count] did not improve or decrease the accuracy. As shown in Table 15, like the star-edges dataset, the [To] feature (or any other feature in addition to the [To] feature) resulted in the lowest accuracy.

To determine the reconnaissance attacks using the non-attack-edges data, average duration and hop count perform the best using the DT algorithm.

Table 14. Decision Tree Results: Non-attack-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Avg_duration, Hop_count	0.985	0.944	0.98	0.984	0.02	0.99	[49, 1 0, 17]
Avg_duration, Hop_count + Any other feature(s)	0.985	0.944	0.98	0.984	0.02	0.99	[49, 1 0, 17]

Table 15. Decision Tree Results: Non-attack-edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
To	0.072	0.208	0	0	1	0.5	[0, 53 0, 14]
To + Any other feature(s)	0.072	0.208	0	0	1	0.5	[0, 53 0, 14]

5.3. Support Vector Machine Results

5.3.1. Star-Edges Dataset

Using SVM, there were over 30 combinations of features that allowed for an accuracy of 100% (1). The minimal number of features required to reach an accuracy of 1 was 2: [From and To], that is Source IP and Destination IP respectively. Table 16 is a truncated table of the SVM Star-edges results. The two unique feature combinations that resulted in an accuracy of 1.0 were [From, To] and [To, Count, Total_Bytes]. The remaining feature combinations were duplicates of these two combinations with additional features and thus, were eliminated from the table to remove redundancy.

Table 16. SVM Results: Truncated Star-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From, To	1	1	1	1	0	1	[78,0 0, 33]
To, Count, Total_Bytes	1	1	1	1	0	1	[83, 0 0, 28]

5.3.2. Non-Attack-Edges Dataset

As presented in Table 17, for the Non-attack-edges dataset, an accuracy of 1 was not reachable. The highest accuracy reached was 0.981, which was reached using the features [Avg_Bytes and Total_Bytes].

Table 17. SVM Results: Non-attack-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Avg_Bytes, Total_Bytes	0.981	1	0.977	0.989	0	0.989	[86, 2 0, 16]
From, Total_Bytes	0.981	1	0.977	0.988	0	0.989	[85, 2 0, 17]
Avg_Bytes, Total_Duration Count, Total_Bytes	0.981	1	0.977	0.988	0.063	0.989	[85, 2 0, 17]
Avg_Bytes	0.981	0.989	0.988	0.989	0.059	0.963	[87, 1 1 , 15]
Count, Total_Bytes	0.981	0.989	0.989	0.989	0.059	0.965	[86, 1 1, 16]
Avg_Bytes, Total_Duration, Avg_Duration, Count	0.981	0.989	0.989	0.989	0	0.965	[86, 1 1, 16]

No worst results table were presented for SVM since most of SVM results were higher on the average.

5.4. Logistic Regression Results

5.4.1. Star-Edges Dataset

Table 18 shows that the [From] feature, that is, the Source IP, was the only feature needed for 100% accuracy using the LR algorithm. Adding additional features did not increase or decrease

the accuracy. The confusion matrix shows that there were no false positives or false negatives. Table 19 presented the worst results for LR using Star-edges.

Table 18. Logistic Regression Results: Star-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From	1	1	1	1	0	1	[78, 0 0, 33]
From, To	1	1	1	1	0	1	[78, 0 0, 33]
From + any other feature(s)	1	1	1	1	0	1	[78, 0 0, 33]

Table 19. Logistic Regression Results: Star-edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Average_Duration	0.63	0	1	0.93	0	0.5	[82, 0 29, 0]
Total_Duration	0.65	0	1	0.94	0	0.66	[84, 0 27, 0]

5.4.2. Non-Attack-Edges Dataset

Table 20 presents results similar to the Star-edges dataset in that the [From] feature, or Source IP, was the only feature needed for 100% accuracy. Adding additional features did not increase or decrease the accuracy. Where these results differ from the previous results is that including only the [Total_duration] feature resulted in the lowest accuracy, as presented in Table 21.

Table 20. Logistic Regression Results: Non-attack-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From	0.983	0.944	0.976	0.980	0.024	0.988	[80, 2 0, 34]
From + any other feature(s)	0.983	0.944	0.976	0.980	0.024	0.988	[80, 2 0, 34]

Table 21. Logistic Regression Results: Non-attack-edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Total_duration	0.705	0.5	0.919	0.886	0.081	0.576	[79, 7 23, 7]

5.5. Gradient Boosting Tree Results

5.5.1. Star-Edges Dataset

Table 22 shows that the [Avg_duration and Hop_count] features resulted in the highest accuracy using the GBT algorithm on the Star-edges dataset. Adding additional features in addition to [Avg_duration and Hop_count] did not increase or decrease the accuracy. The confusion matrix shows that there were two false positives and no false negatives. Table 23 shows that, using the

[To] feature (or any combination of thereof) resulted in the lowest accuracy. The confusion matrix shows a 100% false positive rate for this scenario.

Table 22. Gradient Boost Results: Star-Edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
From	1	1	1	1	01	1	[46, 0 0, 19]
From + any other feature(s)	1	1	1	1	01	1	[46, 0 0, 19]

Table 23. Gradient Boost Results: Star-Edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
To + any other feature(s)	0.072	0.209	0	0	1	0.5	[0, 53 0, 14]

5.5.2. Non-Attack-Edges Dataset

Table 24 shows that the [From] feature, or Source IP, resulted in 100% accuracy using the GBT algorithm. Adding additional features in addition to [From] did not increase or decrease the accuracy. The confusion matrix shows that there were no false positives and no false negatives. Table 25 shows that, using the [To] feature (or any combination of thereof) resulted in the lowest accuracy. The confusion matrix shows a 100% false positive rate for this scenario.

Table 24. Logistic Regression Results: Non-attack-edges Dataset – Best Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
Avg_duration + Hop_count	0.985	0.944	0.98	0.98	0.02	0.99	[49, 1 0, 17]
Avg_duration + Hop_count + any other feature(s)	0.985	0.944	0.98	0.98	0.02	0.99	[49, 1 0, 17]

Table 25. Logistic Regression Results: Non-attack-edges Dataset – Worst Results

Features	Accuracy	Precision	Recall	F-Measure	FPR	AUC-ROC	Confusion Matrix
To + any other feature(s)	0.067	0.2	0	0	1	0.5	[0, 52 0, 13]

5.6. Summary of Results

After analyzing all the results from the individual algorithms, we found that the most important feature that worked best for the Star-edges dataset irrespective of algorithm was the [From] feature, that is, the Source IP. All models that gave perfect performance in terms of accuracy, precision, recall, and F-measure included the [From] feature or Source IP. For the Non-attack-edges dataset, the best results were obtained by LR as well as DT, with an accuracy of 98.5%, both using the features average duration as well as hop_count.

Overall, the models trained with the Star-edges dataset tended to perform better on average than the models trained with the Non-attack-edges dataset. All of the Star-edges models were able to reach a perfect performance in terms of accuracy, precision, recall, and f-measure. Though the Non-attack-edges models, were only able to reach metrics as high as 98.5%, this performance by itself is pretty good overall. Moreover, these results are natural since the Non-attack-edges dataset had some "non-attack" edges mixed in, explaining the slightly lower results.

The best performing algorithm irrespective of dataset appeared to be the DT algorithm. It was able to receive a perfect metric performance for the Star-edges dataset with the minimal number of features, which was the [From] feature or the Source IP. That means, using the Star-edges data, the Source IP itself can predict reconnaissance. The DT and LR algorithms also achieved the best results for the Non-attack-edges dataset, it had the highest accuracy, precision, recall, and F-measure compared to the rest of the algorithms run on the Non-attack-edges dataset.

6. CONCLUSIONS

The objectives of this research were to, first, create ML models capable of detecting Reconnaissance network attacks using graph data from the Zeek Connection Logs Data [28], train the models with various network motif datasets from a graph database, and then finally compare the results. Upon completion of this research, it was concluded that the most important feature for predicting Reconnaissance network attacks within the Zeek Connection Logs dataset was the [From] feature, the Source IP, which represented the network address of where the connection was originating. It was also found that irrespective of which motif was used to train the model, the DT algorithm performed best. These findings provide valuable insights for future research on applying graph database data to machine learners to create models capable of predicting nefarious network traffic.

ACKNOWLEDGEMENTS

This research was funded by the National Centers of Academic Excellence in Cybersecurity, 2021 NCAE-C-002: Cyber Research Innovation Grant Program, Grant Number: H98230-21-1-0170
This research was also supported by the Askew Institute of the University of West Florida.

REFERENCES

[1] S. Morgan, "Cybercrime To Cost The World $10.5 Trillion Annually By 2025," *Cybercrime Magazine*. https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/
[2] Neo4j. What is a Graph Database? Retrieved from https://neo4j.com/what-is-a-graph-database/.
[3] NebulaGraph. (2022, December 8). *Why use a Graph Database? Benefits of Graph Databases*. Open Source Distributed Graph Database. Retrieved April 16, 2023, from https://www.nebula-graph.io/posts/why-use-graph-databases
[4] Zhang, X., Li, J., Zhang, D., Gao, J., & Jiang, H. (2021). Research on Feature Selection for Cyber Attack Detection in Industrial Internet of Things. In Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies (pp. 256-262). Association for Computing Machinery.
[5] Matthew Beechey, Konstantinos G. Kyriakopoulos, & Sangarapillai Lambotharan (2021). Evidential Classification and Feature Selection for Cyber-Threat Hunting. Knowledge-based Systems, 226, 107120.
[6] Gupta, A. (2023, April 26). *Feature selection techniques in Machine Learning (updated 2023)*. Analytics Vidhya. Retrieved May 7, 2023, from https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning

[7] What Is the MITRE ATT&CK Framework? | Get the 101 Guide. (2022). Trellix. https://www.trellix.com/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html (accessed on 1st October, 2023).

[8] MITRE ATT&CK (2023) Reconnaissance, Tactic TA0043 - Enterprise | MITRE ATT&CK®. https://attack.mitre.org/tactics/TA0043/

[9] Prusti, D., Das, D. & Rath, S.K. Credit Card Fraud Detection Technique by Applying Graph Database Model. *Arab J Sci Eng* 46, 1–20 (2021). https://doi.org/10.1007/s13369-021-05682-9

[10] Magomedov, Shamil & Pavelyev, Sergei & Ivanova, Irina & Dobrotvorsky, Alexey & Khrestina, Marina & Yusubaliev, Timur. (2018). Anomaly Detection with Machine Learning and Graph Databases in Fraud Management. International Journal of Advanced Computer Science and Applications. 9. 10.14569/IJACSA.2018.091104.

[11] Petrovic, Nenad. (2021). Booking Cancellation Prediction Relying on Graph Database in Neo4j.

[12] Zhang, X., Feng, Y., & Wang, H. (2021). Identifying influential nodes in social networks using graph databases and machine learning. Journal of Ambient Intelligence and Humanized Computing, 12(3), 2243-2254

[13] C. Coupette and J. Vreeken, "Graph Similarity Description," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021, doi: 10.1145/3447548.3467257.

[14] Sharan, S., et al. (2014). Integrating graph databases and machine learning for drug target discovery. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1396-1405).

[15] Sarah Parisot, Sofia Ira Ktena, Enzo Ferrante, Matthew Lee, Ricardo Guerrero, Ben Glocker, & Daniel Rueckert (2018). Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease. *Medical Image Analysis, 48, 117–130.*

[16] Schindler, T. (2018). Anomaly detection in log data using graph databases and machine learning to defend advanced persistent threats. *arXiv preprint arXiv:1802.00259*.

[17] K. Sotiropoulos, L. Zhao, P. J. Liang, and L. Akoglu, "ADAMM: Anomaly Detection of Attributed Multi-graphs with Metadata: A Unified Neural Network Approach," *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2311.07355.

[18] M.-C. Lee, H. T. Nguyen, D. Berberidis, V. S. Tseng, and L. Akoglu, "GAWD: graph anomaly detection in weighted directed graph databases," *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, vol. 20, pp. 143–150, Nov. 2021, doi: 10.1145/3487351.3488325.

[19] S. S. Bagui, D. Mink, S. C. Bagui, D. H. Sung, and F. Mahmud, "Graphical representation of UWF-ZeekData22 using Memgraph," *Electronics*, vol. 13, no. 6, p. 1015, Mar. 2024, doi: 10.3390/electronics13061015.

[20] Bagui, S. S., Mink, D., Bagui, S. C., Plain, M., Hill, J., Elam, M. (2023). Using a Graph Engine to Visualize the Reconnaissance Tactic of the MITRE ATT&CK Framework from UWF-ZeekData22, *Future Internet*, 15, 236. https:// doi.org/10.3390/fi15070236

[21] Anil. (2020, December 21). *Motifs and structural roles in networks*. W&B. Retrieved April 9, 2023, from https://wandb.ai/syllogismos/machine-learning-with-graphs/reports/3-Motifs-and-Structural-Roles-in-Networks--VmlldzozNzU1NTg

[22] Hu, Jiafeng & Cheng, Reynold & Chang, Kevin & Sankar, Aravind & Fang, Yixiang & Lam, Brian. (2019). Discovering Maximal Motif Cliques in Large Heterogeneous Information Networks. 746-757. 10.1109/ICDE.2019.00072.

[23] Zhang, Z. (2019, August 13). *Naive bayes explained*. Medium. Retrieved April 18, 2023, from https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0

[24] Thorn, J. (2021, September 26). *Decision trees explained*. Medium. Retrieved April 3, 2023, from https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6

[25] Pupale, R. (2019, February 11). *Support vector machines(svm) - an overview*. Medium. Retrieved April 1, 2023, from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

[26] Pant, A. (2019, January 22). *Introduction to logistic regression*. Medium. Retrieved April 3, 2023, from https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[27] Grover, P. (2022, December 13). *Gradient boosting from scratch*. Medium. Retrieved April 3, 2023, from https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

[28] University of West Florida. 2022. Available online: https://datasets.uwf.edu/ (accessed on 20 August 2020).

[29] Bagui, S.S., Mink, D., Bagui, S.C., Ghosh, T., Plenkers, R., McElroy, T., Dulaney, S. & Shabanali, S. (2023). Introducing UWF-ZeekData22: A Comprehensive Network Traffic Dataset Based on the MITRE ATT&CK Framework. *Data*, 2023, **8**, 18. https://doi.org/10.3390/data8010018.

[30] Bagui, S., Mink, D., Bagui, S., Ghosh, T., McElroy, T., Paredes, E., Khasnavis, N., & Plenkers, R. (2022). Detecting Reconnaissance and Discovery Tactics from the MITRE ATT&CK Framework in Zeek Conn Logs Using Spark's Machine Learning in the Big Data Framework, *Sensors*, 22, 7999. https://doi.org/10.3390/s22207999

AUTHORS

Dr. Sikha Bagui is Distinguished University Professor and Askew Fellow in the Department of Computer Science, at The University West Florida, Pensacola, Florida. Dr. Bagui is active in publishing peer reviewed journal articles in the areas of database design, data mining, Big Data analytics, machine learning and AI. Dr. Bagui has worked on funded as well unfunded research projects and has 100+ peer reviewed publications in highly selected journals and conferences. She has also co-authored several books on database and SQL. Bagui also serves as Associate Editor and is on the editorial board of several journals.

Dr. Dustin Mink received the B.S. and M.S. in Computer Science from the University of West Florida, and Ph.D. in Computing from the University of South Alabama. He is Faculty in the Department of Cybersecurity and Information Technology at the University of West Florida. Concurrently, he is a Communications Officer at the Marine Corps Reserve assigned as an Adjunct Professor in the Cyber Intelligence and Data Science in Intelligence Department at National Intelligence University. Concurrently, he is a Solution Architect at Leidos contracted to the Department of Defense. His research interests include cyber and signals artificial intelligence and machine learning.

Dr. Subhash C. Bagui received his B.Sc. in Statistics from University of Calcutta, M. Stat. from Indian Statistical Institute and Ph.D. from University of Alberta, Canada. He is currently a University Distinguished Professor at the University of West Florida. He has authored a book titled, "Handbook of Percentiles of Non -central t-distribution", and published many high quality peer reviewed journal articles. He is currently serving as associate editors/ editorial board members of several statistics journals. His research interests include nonparametric classification and clustering, statistical pattern recognition, machine learning, central limit theorem, and experimental designs. He is also a fellow of American Statistical Association (ASA) and Royal Statistical Society (RSS).

Mr. Jadarius Hill is a graduate student in the Computer Science program at the University of West Florida, where he is pursuing a Master's degree. He received his Bachelor's degree in Chemical Engineering from Rice University in 2020. Jadarius's research interest lies in the intersection of machine learning/artificial intelligence and cloud computing. Specifically, he is interested in developing novel machine learning algorithms that can be efficiently executed on cloud-based platforms. Jadarius has worked on a wide spread of research topics, including designing chemical production processes and developing machine learning models capable of detecting computer network intrusion.

Mr. Farooq Mahmud is a software engineer with over 15 years of experience in software development, specializing in cloud-based solutions, microservices architecture, and backend development using .NET and Azure technologies. Mr. Mahmud recently received his master's degree in computer science from the University of West Florida. Mr. Mahmud has led high-impact projects across industries, including the design and implementation of scalable, high-performance systems for companies like Acuity Brands and Marel. With a strong focus on operational excellence, innovation, and mentorship, he has developed expertise in machine learning, and development of high throughput systems. Farooq actively contributes to fostering technical excellence and building robust, cloud-native applications.

Mr. Michael Plain is an instructor in the Department of Computer Science, at Pensacola Christian College, Pensacola, Florida. Mr. Plain has nearly 35 years of experience in software development, working at multiple fortune 500 companies in his career.