Explainable Machine Learning or AI Using Association Rule Mining

Sikha S. Bagui[1*], Emily Summers[1], Dustin Mink[2], Subhash C. Bagui[3]

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

* Corresponding author. Email: bagui@uwf.edu (S.S.B.); summerse40@gmail.com (E.S.); dmink@uwf.edu (D.M.); sbagui@uwf.edu (S.C.B.)
Manuscript submitted March 1, 2026; accepted March 23, 2026; published June 29, 2026.
doi: 10.18178/JAAI.2026.4.2.125-142

Abstract: In this paper feature selection is performed using Association Rule Mining (ARM), a widely used data mining technique. Association Rule Mining is used as a preprocessing step before machine learning algorithms are applied. Association Rule Mining allows us to not only select the features, but also select the feature values, thus creating a useful feature-value subset that can be used as input for machine learning algorithms. To date, many works have been done on feature selection prior to running machine learning algorithms, but work has not been done on selecting the useful value or range/subset of the feature to be used for better and more efficient machine learning classification. Selecting the useful part of the feature would help in better explaining the machine learning results. This research is conducted using a newly created Cybersecurity dataset, UWF-ZeekData22, labeled as per the MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK) framework. Due to the volume of network data, the Hadoop Distributed File System (HDFS) and Apache Spark were used. The results determined the feature range/value/subset that would be useful in the classification of attack tactics in each machine learning classifier, Decision Trees (DT), Support Vector Machines (SVM), Naïve Bayes (NB), and Random Forest (RF), as well as in all classifiers as a whole, confirming that Association Rule Mining can be useful for explainable machine learning/artificial intelligence and showing inter-feature-value relationships. One of the documented drawbacks of ARM, the generation of too many rules, turned out to be an advantage in this research to help classify rare attacks. That is, in addition to ARM feature-subsets being used for regular explainable AI, ARM's feature-subsets can also be used in explaining rare attacks.

Keywords: explainable Artificial Intelligence (AI), feature selection, cyberattacks, association rule mining, data mining, MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK) framework, frequent pattern mining, feature-values

1. Introduction

Machine Learning (ML) and Artificial Intelligence (AI) are being widely used to build Intrusion Detection Systems (IDS), but the challenge is to comprehend the outcomes of these systems. The lack of comprehension often leads to a lack of trust in these outcomes. Explainable AI (XAI) attempts to explain these outcomes and

increase the trust in these systems. But, to date, most XAI systems are post-hoc explainable methods [1]. In this paper we present a pre-hoc explainable method using rules developed from a widely used data mining technique, Association Rule Mining (ARM) [2, 3].

Association Rule Mining, also known as frequent itemset mining, is a data mining algorithm that extracts knowledge by looking for frequently occurring patterns [2–5] in the data. Frequently derived patterns are termed as strong association rules. Interesting or strong association rules are those rules that apply to a reasonably large number of instances and have a reasonably high accuracy on the instances that they apply to. Though it was first introduced by Agrawal *et al.* [2] in the context of transactional databases, ARM has now been extended to various domains including cybersecurity data [6]. ARM presents its results in the form of rules, and these rules are used to generate feature-subsets.

A lot of work has been done on feature selection prior to running ML or AI algorithms [7–10]. The efficiency of ML/AI algorithms depends on the appropriate selection of features, especially in high dimensional data. Reducing the number of features not only increases the training speed of models but also lowers the complexity of the model, making it easier to understand and improve the performance metrics [11]. These performance metrics can be further improved by selecting the value or range of the feature that is important for classification, rather than using the whole feature.

But, work has not yet been done on figuring out how to select the useful part of a feature for ML/AI. Selecting the useful part of the feature would also help in better explaining classification results. In this paper, feature selection is performed using Association Rule Mining (ARM) [2, 3]. ARM fine-tunes feature selection so that the value/part/range of the feature that is better at achieving a higher ML classification rate can be determined. These fine-tuned results also help us use ARM to explain the features or feature values/ranges used for better classification results, leading to explainable ML/AI.

Hence, the uniqueness of this work is in using ARM as a preprocessing step, for fine-tuned feature value/range selection, prior to using ML/AI algorithms. For any feature, the whole range of values for that feature may not be useful for classification; a smaller range of values may be sufficient, and this can be determined by a using binning process prior to using ARM. These feature value-ranges can also be used for interpretability of ML/AL algorithms as well as showing the inter-feature-value relationships.

This research is performed using a newly created Cybersecurity dataset, UWF-ZeekData22 [12], available [13], labeled as per the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework [14]. The MITRE ATT&CK framework is a globally accessible knowledge base of adversary tactics and techniques used to accomplish specific objectives. This framework has the foundations to be responsive to new threats since the framework is developed from a wide selection of data and features that help form fact-based assumptions on when or how an attack develops. UWF-ZeekData22 [12] is the first work to date that creates a rule-base based on Zeek network connections [15]. The feature subsets generated from the ARM rules with high support and confidence are used by ML classifiers to create a rule-based intrusion detection systems to determine what will give us higher classification accuracy.

Due to the volume of network data, in this work, the Hadoop Distributed File System (HDFS) [16] was used to store the data. HDFS is a scalable, fault tolerant system that allows processing of Big Data by making it available across a cluster of computers [16, 17]. Apache Spark, an open-source framework for Big Data analytics, that sits on top of the Hadoop framework, was used for data processing [18].

One of the documented drawbacks of ARM, the generation of too many association rules [2, 3], was turned into an advantage in this research to help classify rare attacks. Using ARM feature subsets, we were able to identify and explain even rare attacks. Since ARM typically generates a lot of rules, two of our attack types, Credential Access and Privilege Escalation, had very few instances, but since ARM generated a lot of rules, enough to run the machine learners, we were able to successfully classify these attack types and explain even

what would be considered rare attacks. This is a significant observation or contribution of this study.

The rest of this paper is divided into the following sections. Section 2 presents the related works, that is, works related to the three main concepts in this paper, ARM, feature selection and explainable AI; Section 3 presents an overview of association rule mining; Section 4 explains the pre-processing steps in detail; Section 5 presents the Machine Learning algorithms and Spark parameters used; Section 6 presents and discusses the results obtained; and Section 7 presents the conclusion.

2. Related Works

There are a few different concepts covered in this paper: association rule mining, feature selection and Explainable ML/AI. Related works on each of these concepts are covered in separate sub-sections.

2.1. Works Related to Association Rule Mining

ARM is discussed in its' completeness and the different variations of its' implementation in [19]. It discusses some of the problems encountered with association rule mining. It also goes on to explain the availability of usage within different fields and purposes to include website navigation analysis, homeland security, medical diagnosis/research, and market-based analysis. The goal of this research article is to map features in an if/then context. Similar but variably different research is included in [6].

The Adaptive-Miner algorithm was used to construct a rule base for cyber intrusion detection in [20]. The rule base would then be used to conduct security intrusion and vulnerability detection. Additionally, research was conducted to discover how attribution to cyberattacks correlated with actual cyber-attacks. This can improve future cyberattack attributions. Within similar research [21] was conducted where server log files were used to analyze data with association rule mining. This article uses the tree data structure to map association rules and then uses them to predict a client's behavior.

Association rulesets have been generated for rule appliance for cybercriminal detection such as the Cyber Threat Intelligence Association Ruleset (CTI-AR), which were created to be used by security analysts in discovering hidden knowledge from within to help track cybercriminals in the future [21].

Apriori algorithm analysis was used with Defense Advanced Research Projects Agency (DARPA) data from 1999 and 2000. Using a support threshold of 40% and 93%, cyberattacks were immediately detectable within twenty seconds with a real-time scenario of an 84% attack detection [6].

Guyon *et al.* [22] proposed Recursive Feature Elimination (RFE). This was mainly used with the SVM model. Another feature selection algorithm, Boruta [23], has mainly been applied to the Random Forest. Raman and Ioerger [24] suggest a Sequential Forward Select greedy search strategy for feature selection that starts with an empty set and adds features in a greedy manner until the performance can no longer be improved.

None of the previous works used the rules generated from ARM as input for machine learners, which is the gap that this paper is filling. This paper shows how it can be useful to use ARM rules for getting higher classification with machine learners.

2.2. Works Related to Feature Selection

By focusing on only the features that are the most relevant, feature selection helps to build models that are efficient and more reliable as well as easier to interpret when using high-dimensional data [25].

There are five different methods for feature selection: embedded, wrapper, hybrid, ensemble and filter. Pudjihartono *et al.* [9] mentions that the feature selection method is problem-specific and depends on the type of the data being analyzed as well as the goals intended by the researcher.

While feature selection is mainly thought of as a dimensionality reduction tool, it is also used as a means of uncovering intrinsic data structures as shown by Cai *et al.* [7]. Xue *et al.* [10] used it to determine classification accuracy. In the study [5], the use of feature selection in association rule mining was proven to work well,

improving the performance of the Apriori algorithm significantly and thus improving the association rule mining process.

Again, none of the previous works have used association rule mining for feature selection or feature-value selection ,which is the gap that this paper is addressing.

2.3. Works Related to Explainable AI

Feature selection works hand in hand with explainable AI, as shown by Rostami *et al.* [26]. This study showed that using feature selection with decision trees and visualizations allowed clinicians to see which blood tests contribute the most as features. It also showed that reducing the features speeds up the official diagnosis and also helps make the results more explainable.

Ali *et al.* [1] provides a broad literary review of what is known in Explainable Artificial Intelligence (XAI) and highlights the challenges that remain for achieving trustworthy AI, emphasizing that explainability must go hand in hand with ethical, legal, and societal standards. This review goes forth to state that these aspects are crucial to build trust in AI among scientists, the wider public, politicians and regulators.

XAI has its drawbacks, especially when it comes to the data that it is trained on. Incorrect data or biased data can produce explainable work but also enforce discriminatory bias based on objectification [27]. According to Hussain, not much work has been done on setting up regulatory frameworks for AI, which will make XAI a tool for insights rather than a formality.

XAI has two dimensions of interpretability: Local and global. Global interpretability is aimed at understanding the overall decision logic of a ML model and local interpretability is used to understand the specific decisions that are made by that ML model [28]. As found in one study, 512 peer reviewed articles were studied and findings showed that the most common XAI field is medicine where local interpretability is found to be the most important [29]. In this study, most explanations were given from expert opinion and qualitative tests rather than standardized metrics, which is one of the challenges of XAI. In order to reliably say which explanation method works better, quantitative evaluation metrics are necessary.

Lastly, SHAP [30–32,] and LIME [33, 34] are two well-known XAI models that are increasingly being integrated into supervised ML models to gain a deeper understanding of ML predictions. Both these models focus on explainability post hoc, that is, are used to explain ML and AI results.

Though our study has the ability to be used post hoc, the focus of our study is on using it for feature-value selection, that is, using association rule mining feature-values as input to ML and AI models for better results in ML and AI models. This is very necessary in the context of big data, where feature-values can highly reduce the complexity of the ML runs as well as the computation time. And, the context of this work is in doing this through the use of several attack tactic data, labeled as per the MITRE ATT&CK framework.

3. Association Rule Mining

Association Rule Mining, or the Apriori algorithm, is an algorithm used to mine frequent itemsets. First, the set of frequent one-itemsets are found, followed by the set of two itemsets, followed by the set of three itemsets, and so on. At each iteration, using a join operation, the Apriori algorithm generates a set of candidate itemsets from the frequent itemsets found at the previous iteration. Pruning is done by removing, at each iteration, all itemsets that fall below a particular user-defined support (frequency) threshold. The frequency of these candidate itemsets is calculated by scanning the input dataset, resulting in frequent scans of the database. If the frequency count of a candidate itemset is more than the user-defined minimum support, that candidate itemset is considered frequent. An iteration number represents the length of the itemset.

The strength of an association rule is measured in terms of the rule's statistical significance, known as support and confidence. The final output is a set of rules, known as the strong association rules, that satisfy a user-specified minimum support and confidence [2, 3, 5, 35]. Support determines how often a rule occurs

in a given dataset and confidence determines how frequently items in Y appear in transactions that contain X. Simply stated, support s is the percentage of transactions in dataset *D* that contain X ∪ Y, that is, contain both X and Y. Confidence *c* is the percentage of transactions in D containing X that also contain Y. This is taken to be the conditional probability, P(X|Y). Lift computes the ratio between the rule's support and confidence, that is, lift is defined by the probability of *A* and B occurring together divided by the probability of *A* multiplied by the probability of *B*. Hence lift is measured by [5]:

$$Lift(A, B) = P(A \cap B) / (P(A)P(B)) \quad (1)$$

If the resulting lift value is greater than 1, there is a positive relationship between item sets A and *B*, meaning that if *A* goes up *B* will also go up. If the resulting lift value is less than 1, then there is negative relationship between *A* and *B*, meaning that, if *A* goes up, B goes down. If the resulting value is equal to 1, then *A* and *B* are independent [5].

In summary, the support and confidence are used to find the relationships between the feature-value pairs in the tactics and the lift is used to find the inter-relationships between the feature-value pairs within the tactics.

In this work, ARM was developed in a distributed framework, Apache Spark, to be able to cater to Big Data. The data was divided by attack tactic and ARM was used to find associations between features values/ranges within each attack tactic. Hence ARM was used to find the feature values to determine or distinguish a particular attack tactic.

4. The Data

The Zeek Connection (Conn) Log dataset, UWF-ZeekData22 [12], available at [13], labeled as per the MITRE ATT&CK framework [14], was used for this analysis. This dataset, with 9,280,869 attack records and 9,281,599 benign records, was generated using the Cyber Range at The University of West Florida (UWF) [36].

Zeek's Conn log files provide network connecting information between two points. These logs track the protocols and associated information such as IP addresses, durations, two-way bytes, states, packets and tunnel information [15]. The full list of the features of the Conn log files or UWF-ZeekData22 is available at Refs. [12, 13].

4.1. Tactics Studied

For this study, the connections leading to four adversary tactics were studied: Reconnaissance [37], Discovery [38], Credential Access [39], and Privilege Escalation [40]. There were too few instances of the other tactics, hence they were omitted from the study.

4.1.1. The Reconnaissance Tactic

The Reconnaissance tactic is used by adversaries to survey the network and gather information that will enable them to carry out attacks in the future [38]. This dataset, mostly composed of the reconnaissance tactic, has 9,278,722 reconnaissance records, mostly T1595 active scanning tactics [37].

4.1.2. The Discovery Tactic

The discovery tactics are used by adversaries to learn about the network [39]. Discovery has the second highest number of tactics in this dataset. Of the 2086 discovery records in this dataset, most are the T1046 technique, where adversaries attempt to get a list of the services running on remote hosts and the local network [38].

4.1.3. Credential Access

The Credential Access tactic covers attacks that steal credentials such as passwords and account names [39]. The techniques used to access these credentials include credential dumping or keylogging. With access to the right passwords, adversaries can access specific systems which can make the attack close to undetectable. This dataset has 31 Credential Access records, all of which are classified as TA0006 in the MITRE ATT&CK nomenclature [12, 13].

4.1.4. Privilege Escalation

The Privilege Escalation tactic covers attacks that steal advanced permissions on a network or system [40]. In this way, instead of exploring a network with unprivileged access, they can explore areas that require elevated permissions to achieve their objectives. This dataset has 13 Privilege Escalation records, all of which are classified under TA0004 in the MITRE ATT&CK nomenclature [12, 13].

4.2. Features Used

The full list of the features of the Zeek Conn log files in UWF-ZeekData22 [13, 41] is presented in Table 1. Table 1 also lists, in the last column, the features that were used for ARM. 16 features were used. IP addresses and port numbers were not used, hence were removed. That is, features dest_ip_zeek, src_ip_zeek, dest_port, src_port were removed.

Table 1. List of in features UWF-ZeekData22

Feature	Descriptor	Used in ARM
Ts	Timestamp	No
Uid	Unique Identifier	No
proto*	Protocol type - TCP/UDP/ICMP	Yes
service*	Type of Service, Example: http	Yes
conn_state*	Connection State	Yes
local_orig*	Local Origin	Yes
local_resp*	Local Resp	Yes
history*	A log of data communication	Yes
duration**	Duration of attack; range from 4.858 to 34.556	Yes
orig_bytes**	Orig bytes; range in value from 21.42 to 174.24	Yes
orig_ip_bytes**	Origin IP address bytes; range in value from 172.23 to 1001.75	Yes
orig_pkts**	Networking packets; range in value from 2.079 to 13.80	Yes
resp_bytes**	Bytes; range in value from 47.25 to 283.07	Yes
resp_pkts**	Networking Packets; range in value from 1.39 to 9.66	Yes
resp_ip_bytes**	Bytes; range in value from 132.56 to 1060.695	Yes
missed_bytes**	Missed bytes; range in value from 0.0 to infinity	Yes
label_tactic	Attack type descriptor	No
dest_port_zeek	Destination Port	No
src_port_zeek	Source Port	No
dest_ip_zeek	Destination IP	No
src_ip_zeek	Source IP	No
Datetime	Date and Time	No
community_id	Hash of Network Connection Parameters	No

Note: Nominal features are marked with one asterik (*) and continuous features are marked with a double asterik (**).

4.3. Preprocessing

Association rule mining in Big Data is very different. Since Big Data would have too many values per feature, especially continuous values, the main preprocessing that was done was binning. As per nominal values too, if there are too many values with very few frequency counts, it is best to bin the nominal values into fewer categories. Since association rule mining is based on frequency counts, having continuous values in columns (features), especially in Big Data, would not generate association rules, so continuous columns, especially the network data used in this study, would have to be binned into fewer discrete values so that associations can

be found between the features or feature ranges.

4.3.1. Binning

Of the features that were used for this study (Table 1), there were some continuous features (marked with a double asterik (**)), some nominal features (marked with an asterik (*)), IP addresses and port numbers. For these features, preprocessing was mainly done using binning, as per [41], as explained in the following section.

Given that the distribution of data might be different for each tactic, binning should be done separately for each tactic. For the purposes of this study, however, the bins were calculated separately and then averaged, for brevity of explanation's sake.

4.3.1.1. Binning continuous valued features

As per Table 1, there were a number of continuous features (duration, orig_bytes, orig_ip_bytes, orig_pkts, resp_bytes, resp_pkts, rep_ip_bytes, missed_bytes), which were binned as per [41], as explained in the following sections.

For the continuous features, first, trimming was performed, for the purposes of removing null values as well as extreme outliers. Several features in the data exhibited skewed data distributions, often forming long tails skewing to the right and this would partially compensate for skewness of the feature's data. To account for the skewness caused by lengthy and low-lying tails, a 10% trim on the data was used to generate the bins. This means that 80% of the data was used for mean and standard deviation calculations, with 10% of the lowest ranking and highest-ranking values being removed beforehand. After the trimmed mean and trimmed std deviations were calculated, binning would be outlined given the following edges:

$$edge_0 = -\infty$$
$$edge_1 = \mu - 2\sigma$$
$$edge_2 = \mu - \sigma$$
$$edge_3 = \mu$$
$$edge_4 = \mu + \sigma$$
$$edge_5 = \mu + 2\sigma$$
$$edge_6 = \infty$$
$$edges = [edge_0, edge_1, edge_2, edge_3, edge_4, edge_5, edge_6]$$

In order to maintain the desired number of bins in spite of the skewness of the data, the following moving-mean logic was inserted during the establishment of the edges in order to avoid redundant bin ranges where the minimum value of an attribute never drops below 0:

$$if\ min\_val \geq 0$$
$$while\ \mu - 2\sigma < 0$$
$$\mu = \mu + \sigma$$

The resulting bins for the continuous values are presented in Table 2.

Table 2. Binned orig_bytes

Bytes		Binned_value
orig_bytes	0 or NULL	0
	1–272	1
	273–865	2
	866–1457	3
	>1457	4
orig_pkts	0 or NULL	0
	1–3	1
	3–8	2

	8–13	3
	>14	4
orig_ip_bytes	0 or NULL	0
	1–71	1
	72–122	2
	123–173	3
	>174	4
resp_bytes	0 or NULL	0
	1–62	1
	63–171	2
	172–282	3
	>283	4
resp_pkts	0 or NULL	0
	1–2	1
	3–4	2
	5–9	3
	>9	4
resp_ip_bytes	0 or NULL	0
	1–131	1
	132–595	2
	596–1059	3
	>1060	4
missed_bytes	0 or NULL	0
	>1	1

4.3.1.2. Binning nominal features

Table 1 shows the nominal features with one asterik (*). The nominal features contain non-numeric data, but some of these features contain too many unique values. In the ARM algorithm, since such values may not necessarily indicate an intrinsic value difference, binning is necessary. The nominal features are: proto, conn_state, local_orig, local_resp, history and service. Tables 3–6 present the binned nominal values.

Table 3. Binned Protocol Values

Protocols	Binned_value
UDP	1
TCP	2
ICMP	3

Table 4. Binned conn_state Values

Conn_state	Original_value	Binned_value
conn_state	S0	1
conn_state	SF	2
conn_state	SHR	3
conn_state	OTH	4
conn_state	SH	5
conn_state	RSTRH	6
conn_state	REJ	7
conn_state	RSTR	8
conn_state	RSTO	9
conn_state	S1	10
conn_state	S2	11

Table 5. Binned Local_orig_Values and Local_resp Values

Features		Binned_Value
Local_orig	False	1
	True	2
Local_resp	False	1
	True	2

Table 6. Binned Service Values

Service	Binned_Value

dns	1
dhcp	2
http	3
ntp	4
ssl	5
Others	6

5. Experimental Design

The overall experimental design is presented in Fig. 1. As demonstrated in Fig. 1, first, UWF-Zeekdata22, was run through a binning algorithm as presented in [41]. Four groups of data were extracted by the MITRE ATT&CK tactic: Reconnaissance, Discovery, Credential Access and Privilege Escalation. Each of these groups of data were stored separately, by the respective tactics. First, binning was performed on each tactic separately. Then ARM was run on each of these tactics (by tactic). The frequent itemsets were found and the support, confidence and lift values were calculated, and association rules were generated. The rules coming out of these runs (by tactic) were stored separately. The rules that were generated reflected the strong associations within the respective tactics. The rules were then filtered for the best rules, that is, rules with *support > our predefined support, confidence > our predefined confidence* and chosen lift value(s). The feature value/ranges of the strongest rules, by each respective tactic, were used as input to run the binary machine learning classifiers for classification. For each tactic, the datasets were generated using 80% benign data and 20% attack data. 70% of this data was used for training and 30% was used for testing.

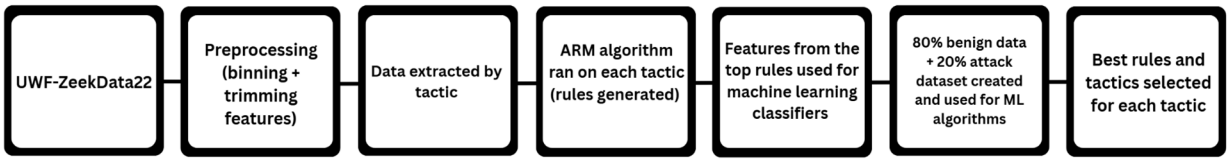

Fig. 1. Experimental design.

5.1. Machine Learning Algorithm and Statistical Metrics Used

The following machine learning algorithms were used for classification: Random Forest (RF), Naïve Bayes (NB), Decision Tree (DT) and Support Vector Machines (SVM). The best rules generated for each attack tactic (by tactic) plus benign data were used as input for each of the machine learners.

5.1.1. The machine learning algorithms used

5.1.1.1. Random forest

The Random Forest (RF) algorithm combines the insights of multiple decision trees to improve decision-making accuracy. In this algorithm, there is a panel of decision trees that come to conclusions and then the majority vote is what constitutes the prediction. This enhances reliability and reduces the chance of error while allowing the handling of complex data to be efficient, providing outcomes even when some data is missing. By using the diverse set of perspectives, RF ensures an accurate analysis and a balanced one, which makes it an ideal choice for a variety of decision making and prediction tasks [42].

5.1.1.2. Naïve bayes

Naïve Bayes (NB) is a straightforward, yet powerful probabilistic classifier grounded in Bayes' Theorem, with the fundamental assumptions that all features in the dataset are independent and contribute equally to the outcome. By calculating the likelihood of each category based on given features and choosing the most

probable one, Naïve Bayes achieves remarkable efficiency and speed, making it well-suited for handling complex, high-dimensional datasets across both categorical and numerical data [43].

5.1.1.3. Decision tree

Decision Tree (DT) is a model that resembles a flowchart. Decision Trees are constituted with nodes that represent features and branches, ultimately representing decision rules with the leaf nodes indicating the outcomes. The algorithm works by starting at the root node and selecting features with the highest information gain. It then continues to sort instances based on the most significant feature at each level - this leads to a classification or outcome [43].

5.1.1.4. Support Vector Machines

Support Vector Machine (SVM) is a model that works to find the hyperplane that best divides a dataset into two classes. The hyperplane is always drawn where it maximizes the margin between the two classes. The data, also called support vectors, are spread across both classes of the hyperplane and those that are positioned closer to the hyperplane help to decide where the hyperplane is drawn [43].

5.1.2. Statistical Metrics

For each algorithm, the Accuracy [11], True Positive Rate (TPR) [44], False Positive Rate (FPR) [44], Precision [44], F-measure [44], and ROC area [45] were determined and recorded. Although this dataset is naturally highly imbalanced, since binary classification was performed, using any almost an equal number of the respective attack tactic and benign data, no measures of dealing with imbalanced datasets were needed. And, although credential access and privilege escalation were small datasets, they generated enough rules that were used as input into the machine learners.

6. Results

A major drawback of association rule mining or frequent pattern mining is that it generates a very large number of frequent itemsets, which reduces the efficiency and effectiveness of the mining results since it is very difficult to sift through a large number of mined rules to find the useful ones [2,3,5, 35]. And many of these rules are also redundant, meaning that there are many subsets that can be found in a superset of the rules. And this phenomenon was applicable to our larger datasets, for the Reconnaissance and Discovery Tactics. But this drawback turned out to be useful for our smaller datasets since we were able to generate enough rules for our smaller datasets, Credential Access and Privilege Escalation, for running our ML algorithms.

The behavior of Big Data is also different from small data in terms of association rule mining. Our initial runs using generally acceptable high support and confidence numbers did not generate enough rules, hence a lot of experimentation had to be done to come up with a significant number of rules to analyze, and different support and confidence values were used for each tactic. Since each tactic was dealt with individually, that is, each tactic would be used for binary classification in the machine learners (that is, one tactic + benign data), the support and confidence for each tactic was determined individually. Moreover, since association rules were going to be determined individually for each tactic, the imbalanced nature of the data did not have to be addressed separately.

For the reconnaissance tactic, we settled for support and confidence greater than 60%; for the discovery tactic, we settled for support and confidence greater and 65%; for credential access, we settled for support and confidence greater and 70% and for privilege escalation, we settled for support and confidence greater than 80%. These support and confidence values generated enough rules for us to be able to use with the machine learners. The smaller datasets had high support and confidence values. Most of these results had lift

value of one or close to 1.0. To prevent overfitting in the machine learners, pruning we performed by removing rules that were strict subsets of higher confidence rules.

To bring the results into perspective, in the following sections, association rules with high accuracy and other statistical measures across all classifiers were presented by attack tactic. The selection criteria for the overall best rules for classification followed an ensemble method. That is, for each tactic, only rules that were common to three for the four classifiers, random forest, naïve bayes, decision trees and support vector machines, were kept and analyzed for explainability. And only the best results of the three of the four classifiers are presented.

6.1. Reconnaissance Results

For the reconnaissance tactic,193 rules were generated with a support greater than 60% and confidence greater than 60%. Using this data, the RF, NB and DT classifiers were run. Table 7 presents the ML classifier results for all the rules with an accuracy greater than 99% across all three classifiers. For brevity's sake, some additional rules are presented in Appendix A of the Supplemental Materials.

Table 7 shows the association rules which have a support close to 95% and confidence close to 99%. From Table 8 it can be noted from rule 33, for example, for the reconnaissance tactic, whenever resp_bytes = 1, orig_bytes = 1, and orig_bytes=1, and this happens with a support greater than 94.1% and confidence of 98.9% Rule. This means that, in this reconnaissance dataset, resp_bytes = 1, orig_bytes = 1, orig_bytes=1 happens 94.% of the time and given orig_bytes=1 and bytes = 1, orig_bytes = 1 happens 98.9% of the time.

Table 7. Reconnaissance: Accuracy ≥ 99% across RF, NB, DT

ARM Rule	Accuracy	TP Rate	FP Rate	Precision	F-Measure	ROC Area
31	99.41%	99.40%	0.80%	99.40%	99.00%	99.30%
33	99.49%	99.50%	0.60%	99.50%	99.00%	99.50%
35	99.41%	99.40%	0.80%	99.40%	99.00%	99.30%
40	99.41%	99.40%	0.80%	99.40%	99.00%	99.30%
44	99.41%	99.40%	0.80%	99.40%	99.00%	99.30%

Table 8. Reconnaissance: Accuracy ≥ 99% across RF, NB, DT

Similar Rules	Rule Description
31	['service=2', 'resp_bytes=1' ⇒ 'orig_bytes=1'] support =0.989, confidence = 0.989, lift = 1.0
33	['proto=1', 'resp_bytes=1' ⇒ 'orig_bytes=1'] support =0.941, confidence = 0.997, lift = 1.008
35	['missed_bytes=1', 'service=2', 'resp_bytes=1'⇒ 'orig_bytes=1'] support =0.989, confidence = 0.989, lift = 1.0
40	['local_orig=1', 'service=2', 'resp_bytes=1' ⇒'orig_bytes=1'] support =0.989, confidence = 0.989, lift = 1.0
44	['local_resp=1', 'service=2', 'resp_bytes=1' ⇒ 'orig_bytes=1'] support =0.989, confidence = 0.989, lift = 1.0

In terms of explainability, as per rule 33, when protocol is UDP (bin 1) and response bytes are between 1 and 62, and original bytes are between 1 and 272, this can classify a reconnaissance attack with 99.49% accuracy.

The inter-feature value-range relationship is shown by the lift values. For example, in rule 31, when [service = 2 (dhncp) and resp_bytes = 1 (between 1 and 62)], then [orig_bytes = 1 (between 1 and 272)] are independent. This would mean that, when service is dhncp, the response bytes and original bytes behave independently. Once does not affect the other.

In rule 33, however, it shows that there is a positive relationship between [protocol = 1 (UDP) and orig_bytes = 1 (between 1 and 272)], and [resp_bytes = 1 (between 1-62)], for the reconnaissance tactic. The positive relationship means, if [proto = 1 and orgn_bytes = 1] goes up, [resp_bytes = 1] goes up. This would mean that, given the protocol as UDP, of the number of original bytes went up, the response bytes would go up.

From Table 7, it can also be noted that, in the reconnaissance tactic, rules with higher support and confidence and lift values of one or greater than one had high accuracy, precision as well as True Positive Rate in all three classifiers, random forest, naïve bayes as well as decision trees. The False Positive Rates were also very low.

6.2. Discovery Results

For the discovery tactic,196 rules were generated with a support greater than 65% and confidence greater than 65%. Table 9 presents the ML classifier results for all the rules with an accuracy greater than 99% across all three classifiers, RF, NB and DT. Table 10 presents the association rules with support and confidence close to 99%, that were common in all three classifiers. For brevity's sake, some additional rules are presented in Appendix B of the Supplemental Materials.

In terms of explainability and inter-feature-value relationships, from Table 10 we note that, for discovery, from rule 32, whenever service = 2 (dhncp) and orig_bytes = 1 (between 1 and 272), then proto = 1 (UDP). Lift, in all cases, was 1.0, meaning that all these features are independent of each other, for the discovery tactic.

From Table 10, it can also be noted that, in the discovery tactic, rules with higher support and confidence and lift values of one had high accuracy, precision as well as True Positive Rate in all three classifiers, random forest, naïve bayes as well as decision trees. For the discovery tactic, however, unlike the reconnaissance tactic, the False Positive Rates were very low.

Table 9. Discovery: Accuracy ≥ 99% across RF, NB, DT

ARM Rule	Accuracy	TP Rate	FP Rate	Precision	F-Measure	ROC Area
32	99.87%	99.90%	0.10%	99.90%	99.90%	99.90%
50	99.87%	99.90%	0.10%	99.90%	99.90%	99.90%
55	99.87%	99.90%	0.10%	99.90%	99.90%	99.90%

Table 10. Discovery: Accuracy ≥ 99% across RF, NB, DT

Similar Rules	Rule Description
32	['service=2', 'orig_bytes=1', 'local_orig=1' ⇒ 'proto=1'] support =0.987, confidence = 0.987, lift = 1.0
50	['service=2', 'orig_bytes=1', 'proto=1' ⇒ 'local_resp=1'] support =0.987, confidence = 0.987, lift = 1.0
55	['service=2', 'orig_bytes=1', 'missed_bytes=1'⇒ 'proto=1'] support =0.987, confidence = 0.987, lift = 1.0

6.3. Credential Access

Although there were only 31 Credential Access instances in this dataset, ARM generated 83 rules with a support greater than 50% and confidence greater than 50%. These were enough rules to run the ML classifiers. Table 11 presents the ML classifier results running RF, NB, and DT, which shows that these features identified the credential access tactic with a 100% accuracy. All other statistical results were also at a 100%. The false positive rate was at zero percent. However, it was difficult to get association rules from the credential access tactic since the best rules were at support of 58% and confidence of 58%. The rules are presented in Table 12. These rules showed a relationship between missed_bytes = 1, proto = 2 and resp_bytes = 2. The lift value of 1.0 showed that these features were independent of each other.

Table 11. Credential Access: Accuracy = 100% across RF, NB, DT

ARM Rule	Accuracy	TP Rate	FP Rate	Precision	F-Measure	ROC Area
31	100.00%	1	0	1	1	1
49	100.00%	1	0	1	1	1

Table 12. Similar Rules Within Credential Access across RF, NB, DT

Similar Rules	Rule Description
31	['missed_bytes=1', 'proto=1', 'resp_bytes=2'⇒ 'local_resp=1'] support =0.580, confidence = 0.580, lift = 1.0
49	['local_orig=1', 'proto=1', 'resp_bytes=2'⇒ 'missed_bytes=1'] support =0.580, confidence = 0.580, lift = 1.0

In credential access, since strong rules were not found using the RF, NB and DT classifiers, the SVM classifier was run and results compared with DT and RF. These results are presented in Table 13, and the rules are presented in Table 14. From Table 14 it can be noted that proto=1 is in every rule. Proto = 1 and local_resp = 1 gave a support of 100% as well as a confidence of 100%. For brevity's sake, some additional rules for credential access are presented in Appendix C of the Supplemental Materials.

In terms of explainability and inter-feature-value relationships, from Table 12, it can be noted that, in credential access, rules with only about 58% support and confidence gave high accuracy using RF, NB and DTs. The False Positive Rate was also low for these two rules. This means that, although, for example, missed_bytes=1, proto=1 (UDP), resp_bytes=2 (between 63 and 171) => local_resp = 1 (False, that is, no response), occurred only 58% of the time in the dataset, this combination of feature subsets were enough to classify the credential access tactic successfully using the random forest, naïve bayes, and decision tree classifier.

From Table 14, it can be noted that, when the SVM classifier was used, some additional rules were added with higher support and confidence. Except for rules 13 and 22, the other rules had higher support and confidence values. Rule 41 followed the same trend as the reconnaissance and discovery tactics, achieving high accuracy and precision with high support and confidence values, that is support and confidence at 100%. The False Positive Rates were at zero in all cases.

From rule 41 (Table 14) it can be inferred that, if protocol = 1 (UDP), then original bytes is 1, that is, between 273 and 865 bytes, and that protocol of UDP and original bytes =1 are independent of one another (shown by the lift value of 1.0). That is, if one goes up, the other does not necessarily go up in the credential access tactic.

Table 13. Credential Access: Accuracy = 100% across Classifiers SVM, RF, DT

ARM Rule	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
13	100.00%	1	0	1	1	1	1
21	100.00%	1	0	1	1	1	1
22	100.00%	1	0	1	1	1	1
37	100.00%	1	0	1	1	1	1
41	100.00%	1	0	1	1	1	1
42	100.00%	1	0	1	1	1	1
44	100.00%	1	0	1	1	1	1
48	100.00%	1	0	1	1	1	1

Table 14. Credential Access: Similar Rules across DT, SVM and RF

Similar Rules	Rule Description
13	['missed_bytes=1', 'proto=1', 'resp_bytes=2'⇒ 'local_resp=1'] support =0.580, confidence = 0.580, lift = 1.0
21	['local_orig=1', 'proto=1', 'resp_bytes=2'⇒ 'missed_bytes=1'] support =0.580, confidence = 0.580, lift = 1.0
22	['proto=1'⇒ 'orig_bytes=2'] support =0.741, confidence = 0.741, lift = 1.0
37	['missed_bytes=1', 'proto=1', 'orig_bytes=2'⇒ 'local_resp=1'] support =0.741, confidence = 0.741, lift = 1.0
41	['missed_bytes=1', 'proto=1', 'local_resp=1'⇒'local_orig=1'] support =1.0, confidence = 1.0, lift = 1.0
42	['missed_bytes=1', 'proto=1⇒ 'orig_bytes=2'] support =0.741, confidence = 0.741, lift = 1.0
44	['missed_bytes=1', 'proto=1', 'orig_bytes=2'⇒ 'local_orig=1'] support =0.741, confidence = 0.741, lift = 1.0
48	['proto=1', 'orig_bytes=2'⇒ 'local_resp=1'] support =0.741, confidence = 0.741, lift = 1.0

6.4. Privilege Escalation

Although there were only 13 instances of Privilege Escalation in this dataset, ARM generated 193 rules with a support greater than 80% and confidence greater than 80%. This was enough to successfully run the ML classifiers. Running the NB, RF and DT classifiers, most rules in privilege escalation had the same results, as shown in Table 15, and the respective association rules are presented in Table 16. Additional rules for privilege escalation are presented in Appendix D of the Supplemental Materials. In Tables 15 and 16 respectively, the rules are presented from 26–50, since rules 1–25 are presented in Appendix D of the Supplemental Materials.

In privilege escalation, the rules with the highest support and confidence values did not generate the highest accuracy. This means that, although these feature-range combinations occur many times within the privilege escalation dataset, they were not the best in classifying the privilege escalation tactic. Overall, the classification accuracy, precision, recall, F-measure as well as ROC Area were also lower for the privilege escalation tactic than the other three tactics studied. The False Positives Rates were also slightly higher than the other tactics studied. This was also the smallest dataset, so this might be something that will have to be studied further.

In terms of explainability and inter-feature-value relationships, we will analyze rule 28. As per rule 28, when the orig_ip_bytes=2 (that is, between 72 and 122), resp_pkts=2 (that is, between 63–171), orig_pkts=2 (that is, between 3 and 8), then local_orig=1 (that is, false), and this happens 74.3% of the time in the privilege escalation tactic (shown by support of 0.743). And, given [local_orig=1 (that is, false)], the [orig_ip_bytes=2 (between 72 and 122), resp_pkts=2 (between 63 and 171), and orig_pkts=2 (between 3 and 8)] happens 100% of the time in privilege escalation (shown by confidence = 1.0). The lift value greater than one implies that when the [orig_ip_bytes=2 (between 72 and 122), resp_pkts=2 (between 63 and 171), and orig_pkts=2 (between 3 and 8)] increases, local_origin also goes up.

Table 15. Privilege Escalation: Accuracy ≥ 92.3% across NB, RF, DT

ARM Rule	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
26	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
27	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
28	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
30	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
31	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
32	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
35	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
37	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
38	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
39	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
40	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
42	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
43	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
44	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
45	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
46	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
47	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%
50	92.30%	92.3%	9%	93.3%	92.3%	92.2%	91.7%

Table 16. Privilege Escalation: Accuracy ≥ 92.3% across RF, NB, DT

Rules	Rule Description
26	['missed_bytes=1', 'orig_pkts=2', 'local_orig=1' ⇒'resp_ip_bytes=2'] support =0.743, confidence = 0.743, lift = 1.0
27	['resp_ip_bytes=2', 'orig_pkts=2', 'local_orig=1'⇒'local_resp=1'] support =0.743, confidence = 1.0, lift = 1.344
28	['orig_ip_bytes=2', 'resp_pkts=2', 'orig_pkts=2'⇒ 'local_orig=1'] support =0.743, confidence = 1.0, lift = 1.344

30	['missed_bytes=1', 'orig_ip_bytes=2', 'orig_pkts=2'⇒'local_resp=1'] support =0.743, confidence = 0.743, lift = 1.0
31	['resp_pkts=2', 'orig_ip_bytes=2'⇒'local_resp=1'] support =0.743, confidence = 1.0, lift = 1.344
32	['resp_pkts=2', 'orig_ip_bytes=2'⇒ 'resp_ip_bytes=2'] support =0.743, confidence = 1.0, lift = 1.344
35	['orig_ip_bytes=2', 'orig_pkts=2'⇒'local_orig=1'] support =0.743, confidence = 1.0, lift = 1.344
37	['orig_ip_bytes=2', 'orig_pkts=2'⇒'local_resp=1'] support =0.743, confidence = 1.0, lift = 1.344
38	['resp_pkts=2', 'orig_ip_bytes=2'⇒'orig_pkts=2'] support =0.743, confidence = 1.0, lift = 1.344
39	['resp_ip_bytes=2', 'local_orig=1'⇒ 'local_resp=1'] support =0.743, confidence = 1.0, lift = 1.0
40	['resp_pkts=2', 'missed_bytes=1'⇒'orig_ip_bytes=2'] support =0.743, confidence = 1.0, lift = 1.344
42	['resp_pkts=2', 'orig_pkts=2', 'orig_ip_bytes=2'⇒'local_resp=1'] support =0.743, confidence = 1.0, lift = 1.344
43	['resp_pkts=2', 'orig_pkts=2'⇒ 'local_orig=1'] support =0.743, confidence = 1.0, lift = 1.344
44	['missed_bytes=1', 'resp_pkts=2', 'orig_pkts=2' ⇒'resp_ip_bytes=2'] support =0.743, confidence = 0.743, lift = 1.0
45	['missed_bytes=1', 'orig_ip_bytes=2', 'local_orig=1'⇒ 'resp_ip_bytes=2'] support =0.743, confidence = 0.743, lift = 1.0
46	['orig_ip_bytes=2', 'orig_pkts=2', 'local_orig=1'⇒ 'resp_ip_bytes=2'] support =0.743, confidence = 1.0, lift = 1.344
47	['resp_pkts=2', 'orig_ip_bytes=2', 'orig_pkts=2'⇒ 'resp_ip_bytes=2'] support =0.743, confidence = 1.0, lift = 1.344
50	['missed_bytes=1', 'resp_pkts=2', 'resp_ip_bytes=2'⇒'local_resp=1'] support =0.743, confidence = 0.743, lift = 1.0

7. Conclusions

Table 17 presents a summary of the results, that is, the similar rules that match up in three classifiers. It can be noted that missed_bytes = 1, local_orig = 1 (that is, false) came up in all classifiers in every tactic (highlighted in yellow). This means that these feature-values or feature-subsets are extremely important in classifying all four tactics. Proto=1 (UDP) and local_resp = 1 came up in three out of the four tactics (highlighted in green). This means that these feature-values or features subsets are useful in classifying three of the four tactics.

Moreover, it can be noted from the high accuracy association rules for most tactics (reconnaissance, discovery as well as credential access), most of the lift values are 1.0. This suggests that high accuracy is driven by feature-value selection in general, and not so much the associative relationship between the feature-values. In privilege escalation, though, there were some high accuracy association rules with lift values greater than one, indicating an associative relationship between the feature-values or feature-subsets in privilege escalation.

Table 17. Features Found in Similar Rules Based on Attack Type

Discovery Compared Across DT, RF, NB	Reconnaissance Compared Across DT, RF, NB	Credential Access Compared Across DT, RF, NB	Credential Access Compared Across DT, RF, SVM	Privilege Escalation Compared Across DT, RF, NB
Service = 2	Service = 2	missed_bytes = 1	local_orig = 1	resp_pkts = 2
orig_bytes = 1	resp_bytes = 1	Proto = 1	Proto = 1	local_orig= 1
local_orig = 1	Proto = 1	resp_bytes = 2	local_resp = 1	missed_bytes = 1
missed_bytes = 1	orig_bytes = 1	local_resp = 1	orig_bytes = 2	orig_ip_bytes = 2
Proto = 1	missed_bytes = 1	local_orig = 1	Duration = 2	local_resp = 1
local_resp = 1	local_orig = 1		missed_bytes = 1	orig_pkts = 2
			conn_state = 1	resp_ip_bytes = 2

In summary, the results support the fact that ARM can be considered useful for explainable ML/AI. It gives a better interpretability of the ML/AI results for each tactic. ARM allows us to find the value of the features that lead to higher classification of each tactic. So, for example, rather than using the protocol feature and the local response bytes feature, using the protocol of 1 (UDP) and local_resp = 1 (false) respectively, will give us better classification results for these tactics.

To present a specific example of explainability, from the reconnaissance tactic, as per rule 33, when protocol is UDP (bin 1) and response bytes are between 1 and 62, and original bytes are between 1 and 272, this can classify a reconnaissance attack with 99.49% accuracy. We would use the values of the features rather than using the whole features, protocol, response bytes and original bytes. In terms of inter-feature-value-range relationships, since lift is greater than 1 in rule 33, there is a positive relationship between [protocol = 1 (UDP) and orig_bytes = 1 (between 1 and 272)], and [resp_bytes = 1 (between 1–62)], for the reconnaissance tactic. The positive relationship means, if [proto = 1 and orgn_bytes = 1] goes up, [resp_bytes = 1] goes up. This would mean that, given the protocol as UDP, if the number of original bytes go up, the response bytes would go up.

On the average, the ML classification results as well as support and confidence values were higher for the reconnaissance and discovery tactics, which means that not only were more feature-value ranges co-occuring, but these co-occurrences were able to classify the respective tactics. For credential access and privilege escalation however, the support and confidence values were lower on the average and though credential access had higher classification results, privilege escalation had lower classification results, the latter indicating that the feature-ranges determined by the association rules were not as strong in terms of classification.

But overall, this study shows how association rule mining can be effectively used for determining feature-value ranges and can be used effectively for classification. The feature-value ranges can also be used for explainability and inter-feature-value relationships. This study also shows how ARM can be successfully used to generate enough rules for classification, even for smaller datasets.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Sikha S. Bagui (SSB), Dustin Mink (DM) and Subhash C. Bagui (SCB) formulated the research problem and binning strategies; Emily Summers (ES) did the programming for the research and collected the results; ES wrote the initial draft of the paper; analysis and validation of the results was done by all authors; SSB, DM and SCB worked on writing, reviewing the editing future drafts of the paper; all authors have approved the final version of the paper.

Acknowledgment

This work was partially supported by the Askew Institute at The University of West Florida.

References

[1] Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what remains. *Information Fusion*, *99*, 101805.

[2] Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data* (pp. 207–216). ACM Press.

[3] Han, J., Pei, J., & Kamber, M. (2012). *Data Mining: Concepts and Techniques* (3rd ed.). Elsevier.

[4] Aggarwal, C. C., Bhuiyan, M. A., & Al Hasan, M. (2014). Frequent pattern mining algorithms: A survey. In C. C. Aggarwal & J. Han (Eds.), *Frequent Pattern Mining* (pp. 19–64). Springer.

[5] Bagui, S., Just, J., & Bagui, S. (2009). Deriving strong association mining rules using a dependency criteria,

the lift measure. *International Journal of Data Analysis Techniques and Strategies*, *1(3)*, 297–312.

[6] Lou, P., Lu, G., Jiang, X., Xiao, Z., Hu, J., & Yan, J. (2020). Cyber intrusion detection through association rule mining on multi-source logs. *Applied Intelligence*, *51(6)*, 4043–4057.

[7] Cai, J., Luo, J., Wang, S., & Yang, S. (2017). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79.

[8] Rajeswari, K. (2015). Feature selection by mining optimized association rules based on Apriori algorithm. *International Journal of Computer Applications, 119(20)*, 30–34.

[9] Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., & O'Sullivan, J. M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics, 2*, 927312.

[10] Xue, Y., Tang, Y., Xu, X., Liang, J., & Neri, F. (2020). Multi-objective feature selection with missing data in classification. arXiv preprint, arXiv:2002.06842.

[11] Özdemir, Ö., & Yıldız, O. T. (2022). A comprehensive review of feature selection and feature selection stability. *Gazi University Journal of Science, 36(4)*, 1412–1446.

[12] 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*, *8(1)*, 18. https://doi.org/10.3390/data8010018

[13] UWF Datasets. Retrieved from https://datasets.uwf.edu/

[14] What Is the MITRE ATT&CK Framework? | Get the 101 Guide. (2024). Trellix. Retrieved from https://www.trellix.com/en-us/security-awareness/cybersecurity/what-is-mitre-attack-framework.html

[15] Zeek Documentation. Retrieved from https://docs.zeek.org/en/master/logs/conn.html

[16] Bagui, S., & Spratlin, S. (2018). A review of data mining algorithms on Hadoop's MapReduce. *International Journal of Data Science, 3(2)*, 146–169.

[17] Apache Hadoop. Retrieved from https://hadoop.apache.org/

[18] Apache Spark (2025). *Spark 4.0.0 Configuration.* Retrieved from https://spark.apache.org/

[19] Kotsiantis, S., & Kanellopoulos, D. (2006). Association rules mining: A recent overview. GESTS *International Transactions on Computer Science and Engineering, 32*, 71–82.

[20] Abu, M. S., Rahayu, S., Yusof, R., & Ariffin, A. (2020). Attribution of cyberattack using association rule mining. *IJACSA, 11(2)*.

[21] Mironeanu, C., Archip, A., & Atomei, G. (2021). Application of association rule mining in preventing cyberattacks. *Bulletin of the Polytechnic Institute of IAŞI, 67(4)*, 25–41.

[22] Guyon, I., Weston, J., Barnhill, S., *et al.* (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*, 389–422.

[23] Kursa, M. B., & Rudnicki, W. R. (2010). Feature selection with the Boruta package. *Journal of Statistical Software, 36(11)*. doi: 10.18637/jss.v036.i11

[24] Raman, B., & Ioerger, T. R. (2002). Instance-based filter for feature selection. *Journal of Machine Learning Research, 1(3)*, 1-23.

[25] Chen, R.-C., Dewi, D., Huang, S.-W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data, 7*, 52.

[26] Rostami, M., & Oussalah, M. (2022). Explainable COVID-19 diagnosis using feature selection and random forest. *Informatics in Medicine Unlocked, 30,* 100941. https://doi.org/10.1016/j.imu.2022.100941

[27] Hussain, A., & Hussain, A. (2025). Transparency and accountability: unpacking the real problems of explainable AI. *AI & Soc.* https://doi.org/10.1007/s00146-025-02302-0

[28] Bassan, S., Amir, G., & Katz, G. (2024). Local vs. global interpretability: A computational complexity perspective. *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)* (Vol. 235).

[29] Saarela, M., & Podgorelec, V. (2024). Recent applications of Explainable AI (XAI): A systematic literature review. *Applied Sciences, 14(19)*, 8884. https://doi.org/10.3390/app14198884

[30] Ponce-Bobadilla, A. V., Schmitt, V., Maier, C., Mensing, S., & Stodtmann, S. (2024). Practical guide to SHAP analysis. *Clinical and Translational Science*, 17*(11)*, e70056.

[31] Hamilton, R. I., & Papadopoulos, P. N. (2024). Using SHAP values to understand transient stability limits. *IEEE Transactions on Power Systems*, *39(1)*, 1384–1397.

[32] Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of machine learning models using SHAP. *Computer Methods and Programs in Biomedicine*, *214,* 106584.

[33] Alodibat, S., Ahmad, A., & Azzeh, M. (2023). Explainable ML-based cybersecurity detection using LIME. *Proceedings of IEEE JEEIT* (pp. 235–242).

[34] Hermosilla, P., Berríos, S., & Allende-Cid, H. (2025). SHAP vs LIME for intrusion detection. *Applied Sciences*, *15*, 7329.

[35] Bagui, S., Just, J., Bagui, S., & Hemasinha, R. (2010). Cosine-type measure for strong association rules. *IJKEDM*, *1(1)*, 69–83.

[36] Miller, E., Mink, D., Spellings, P., Bagui, S. S., & Bagui, S. C. (2025). Classifying cyber ranges. *Encyclopedia*, *5*, 162.

[37] MITRE ATT&CK. *Reconnaissance Tactic (TA0043)*. Retrieved from https://attack.mitre.org/tactics/TA0043/

[38] MITRE ATT&CK. *Discovery Tactic (TA0007)*. Retrieved from https://attack.mitre.org/tactics/TA0007/

[39] MITRE ATT&CK. *Credential Access Tactic (TA0006)*. Retrieved from https://attack.mitre.org/tactics/TA0006/

[40] MITRE ATT&CK. Privilege *Escalation Tactic (TA0004)*. Retrieved from https://attack.mitre.org/tactics/TA0004/

[41] Bagui, S., Mink, D., Bagui, S., Ghosh, T., McElroy, T., Paredes, E., Khasnavis, N., & Plenkers, R. (2022). Detecting reconnaissance and discovery tactics in Zeek logs. *Sensors*, *22*, 7999.

[42] Breiman, L. (2001). Random forests. *Machine Learning*, *45(1)*.

[43] Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Addison Wesley.

[44] Scikit-learn. *Accuracy Score Documentation*. Retrieved from https://scikit-learn.org/

[45] Powers, D. M. W. (2011). Evaluation: From precision, recall and F-Measure. *Journal of Machine Learning Technologies*, *2(1)*, 37–63.