Journal of Theoretical and Applied Information Technology
15th March 2019. Vol.97. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

USING ASSOCIATION RULE LEARNING TO DETECT DDOS ATTACK IN SPV

[1]MARWAN A. ALBAHAR , [1,2]KHALED G. MOHAMED

[1]Department of management information systems, Ibnrushd College for management sciences, Saudi Arabia

[2]Department of mathematics, Faculty of science, Benha University, Egypt

Email: [1]marwan.albahar@ibnrushd.edu.sa, [1,2]kgmohamed@fsc.bu.edu.eg

ABSTRACT

Lightweight Bitcoin clients can authenticate that a transaction is included in the blockchain, without the need of downloading the complete blockchain through Simple Payment Verification (SPV). A vast majority of lightweight clients utilizes SPV. However, with the increase in the popularity of SPV, many attacks have been launched against it such as Spoof, Sniff, Distributed Denial of Service (DDoS). As SPV requires high processing speed, flexibility, and stability, association rule learning can be effectively used for attack detection in SPV. In this paper, we implement a DDoS attack detection system using Association rule learning in SPV. Then, we used a KDD dataset to analyze the detecting accuracy and processing time in comparison with a machine learning approach. In addition, we also used NSL-KDD and GureKDD datasets to validate the effectiveness of our method. Our results show that Association rule learning algorithm is capable of detecting the DDoS attack in SPV.

Keywords: *DDos Detection, Association Rules, Cyber Security, Data Mining, SPV*

1. INTRODUCTION

Nowadays with the advancement of information and communication technologies, we are receiving many conveniences. In addition, as computerized services, as well as existing infrastructures and services, are fused with information and communication technologies, we are once again at the forefront of technological leaps and environmental changes. Based on this information technology, Blockchain technology has been developed and because of this a number of encryption coins have been created that are similar to Bitcoin and a SPV system is widely used to simplify this transaction. In SPV, clients download block headers during the preliminary synchronization procedure and then make transactions from the complete node as required. This spreads linearly, Blockchain heights of 80 bytes per block header or up to 4.2 MB, regardless of the overall size of the block. So, the block headers Merkle root and Merkle branch can verify to the SPV client that the transaction is contained within a block in the Blockchain. This, however, does not warrant the legitimacy of the contained transaction. The blocks depth links to the block of collective difficulty that has been formed over that specific block. The SPV client has the knowledge of the Merkle root and related transaction data and demands the particular Merkle branch from a full node. When the Merkle branch has been reclaimed, verifying the presence of the transaction in the block, the SPV client can view the block depth as an alternative for transaction security and legitimacy. The rate of the DDoS attack on a client by a malicious node, which adds an invalid transaction, increases with a collective difficulty formed over the block since the malicious node will be singly mining this bogus chain [1], [2]. As the implementation of SPV needs to be robust enough to resist different attacks. However, there is a potential weakness in its implementation so a SPV client may experience significant drawbacks.

First, an adversary can't fool an SPV client into thinking it is in a block, not a transaction, but the opposite is not true. The full node may merely recognize that an omission caused by the SPV client not to have been a transaction. This scenario possibly can constitute another form of DDoS attacks. Connecting to multiple nodes and initializing requests to each node can perhaps mitigate it. However, this strategy consumes a lot of bandwidth, because identity is inherently free, and network partitioning or DDoS attacks can overwhelm it. Second, the SPV client only demands transactions on the complete node that link to the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

keys it possesses. If the SPV client downloads all blocks and discards unnecessary packets, this can consume a lot of bandwidth. Consequently, when a user requests a block with a particular node and a specific transaction, the entire node can fully see the public address for that user. This is a significant privacy leak, which enables the owner of the full node to execute techniques like denial of service (DoS) to disfavor the users, clients, or addresses, and it also allows trivial connections of funds. Furthermore, a client can simply send a spoofed transaction request such as spam, but it can be a heavy burden on the SPV client and can ultimately defeat the Bitcoin thin client. As for the earlier reason, Bloom filters have been implemented to reduce bandwidth consumption, and compress the block data requests. A Bloom filter is a data structure designed to tell you whether an element is present in a set, rapidly and memory-efficiently [3]. The purpose of a Bloom filter is to test the membership of an element. All the Bitcoin addresses which appear in the wallet are embedded by an SPV client to construct a Bloom filter then are outsourced to full Bitcoin node after an initial handshake protocol during any communication between an SPV client and the full node. Upon receiving the transaction from the SPV client, the full node checks the similarity of received addresses to the original SPV clients Bloom filter. Based on that checking, the full node is either forwarding a transaction or not to the client (sees Figure 1). [1]– [3]

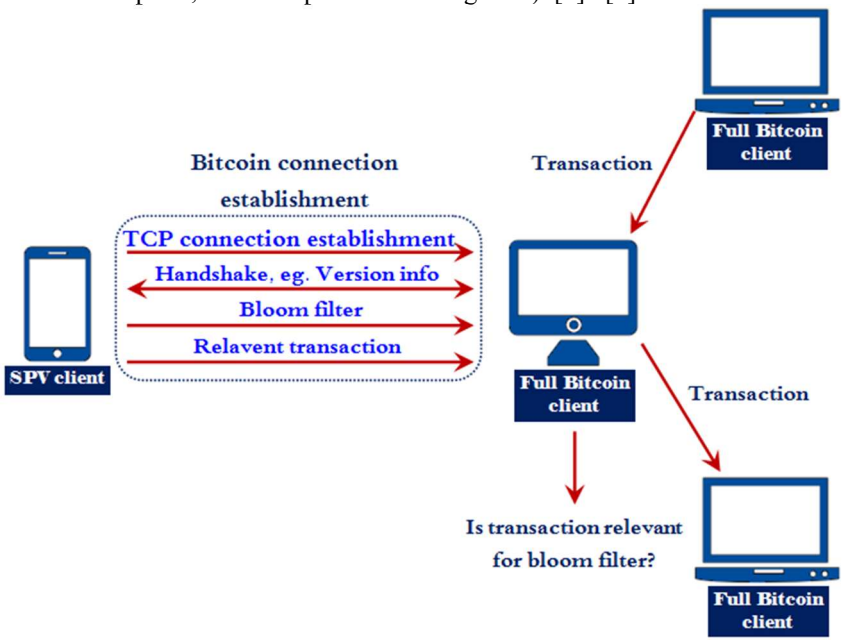


Figure 1. Sketch of SPV client connects to a full Bitcoin node.

The development of information and communication technology does not only have such a pure function. There are a lot of cases in which the defects of information and communication technologies or their environment are exploited to take unfair advantage or harm others.

DDoS is a classic example. Presently, the attack is not just a simple denial of service attack but rather a quite disruptive powerful attack to gain control of computers. Critical resources on the targets are consumed by distributed multiple agents in a short time span whereupon there is a denial of service to authentic clients. After which, network congestion takes place between the source and destination consequently resulting in halting of normal internet processes and denial of service to authentic clients. DDoS is an attack on the accessibility for services of resources of a targeted system that is initiated indirectly from compromised computers known as Zombies on the Internet in a highly coordinated manner. An attacker can utilize client/server technology to significantly increase the effectiveness of a platform attack by using a large number of coordinated zombie resources for it. Zombies perform real attacks by significantly increasing traffic to a victim's computer. Consequently, the target PC loses all communication and computing resources [4], [5] In addition, these attacks are also widely used for monetary gain by influencing the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

top search term modifiers of Internet portal services. In fact, most of the DDoS attacks are aimed at multinational organizations because of the potential of substantial monetary gains. Many new organizations including BitQuick and CoinWallet were out of business within a very short period after their introduction to the market because of such DDoS attacks. A DDoS attack undertakes different forms that discourage the miner, so they pull out of the mining process [6]. The unstructured P2P network in the Bitcoin allows prompt distribution of the data in all parts of the network. The global state of Blockchain governs the safety of Bitcoin depends on the effectiveness of a PoW-based consensus protocol. A consensus protocol can be strongly affected by the differences among the propagation methods. Inconsistent and unreliable Blockchain states could be harnessed to carry out double spending transactions. So, it is integral for the Bitcoin network to stay accessible with respect to network size, network bandwidth, and storage demand as it would help in the growth of the amount of authentic miners present within the network which would make the consensus protocol stronger. As for the most reliable approach in Bitcoin, full nodes start the process from the genesis block to download and authenticate all blocks. A P2P network also involves full nodes participation for the propagation of the information. However, as for the thin clients, the preferable option would be the simplified payment verification (SPV) to carry out Bitcoin transactions even though it has its flaws which expose the client to privacy leaks and DoS attack [7], [8]. With the dramatic expansion of the scope of DDos attack, machine learning is widely used to detect such attacks, but this method requires sufficient datasets and resources and has a long processing time. Detection of service denial attacks by machine learning from these defects cannot be used for SPV systems that must guarantee very fast transactions.

To resolve these issues, we propose a service denial attack detection method that uses Association rule learning algorithm. The proposed method detects a DDoS attack by analyzing the relationships among received packets, in which the connection logs are divided continuously into several groups depending on their time stamps, and those in the same group form a transaction. Then, a DDoS attack detection operation can be performed by a transaction.

Our contributions are as follows:

1. We design a DDoS detection model to protect the lightweight Bitcoin clients based on Association rule learning. As SPV systems must guarantee quick transactions, our model demands fewer resources and has a less processing time.
2. We compare our model to the traditional machine learning model. Also, we show that our model is more efficient in terms of detecting DDoS attacks than machine learning model.

The composition of this paper is as follows. In Section 2, we discuss the association rule learning and its advantages. In Section 3, we describe the DDoS attacks and their detection methods. In Section 4, we define our threat model, and then we explain our proposed method. In Section 5, we provide our experimental results to verify the proposed method efficiency, and then we analyze the effectiveness of the method on different datasets in Section 6. Finally, we conclude our work in Section 7.

2. ASSOCIATION RULE LEARNING (ARL)

Agrawal, Hnielinski and Swami proposed the concept of the association rule that discovers relationships between variables in large databases and generates significant association rules. The purpose of this procedure is to discern valuable linked information among the data description of a large data set. To reflect the appearance of an object and the way it impacts some other object's appearance with the help of a quantification digit, suppose $I = \{i_1, i_2, i_3, \dots, i_m\}$ is termed as a set of collection of binary characters that are known as items. Define T as an items set, and $T \subset I$; define D as a collection of sets from the power set of I; i.e $T \in D$. Define χ as a subset of I, if $\chi \subset T$, then T contains χ. However, as the length of the item set is determined by the numbers of items included in the set, the total number of items included in the set identifies the length of item set. The shape like $\chi \rightarrow \eta$, satisfies $\chi \subset T$, $T \subset I$, and $\chi \cap \eta = \emptyset$.
The support and the confidence formula are:

$$Support\ (\chi \rightarrow \eta) = |\{T : \chi \cup \eta \subseteq T \text{ and } T \in D\}| / |D|,$$

$$Confidence\ (\chi \rightarrow \eta) = |\{T : \chi \cup \eta \subseteq T \text{ and } T \in D\}| / |\{T : \chi \subseteq T \text{ and } T \in D\}|.$$

The objective is to develop an association rule from identified T sets (between the data items set), guaranteeing that its support/confidence is greater than the smallest of the item that the user already had in advance. As a common rule, it differentiates that all the supports are not measured as lower than smallest support set which has been provided by the user (frequent item set). Moreover, the set facilitates in creating a strong connection rule. The complete implementation is based on the first step,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the core step, so this association rule is very simple to achieve when all frequent item sets have been found [9].

3. RELATED WORK

3.1 DDoS Attack

During DDoS attacks, an attacker integrates DDoS traffic from various source points to cause congestion to its target [10]. Typically, DDoS attack sources are widespread worldwide, and they generally vary from hundreds to thousands or sometimes in thousands, and they are often globally distributed. MULTOPS [11] and LADS [12] are traditional DDoS attack detection techniques where they measure the traffic volumes. According to these particular techniques, the identification of DDoS attacks is relied on the measurement of the traffic volume against a specific value of the threshold. Broadly speaking, DDoS attacks can be categorized into two groups that are the low-intensity attacks and the high-intensity attacks. The key difference between these two types of attacks is just the packet transmission rate in the traffic data.

3.2 Blockchain vulnerabilities

In [3], the authors examined the state of existing SPV clients' privacy. They demonstrated through empirical testing the dependence on Bloom filters in the current SPV clients' leaks. The analytical results also verified that SPV clients who employ an unassertive number of Bitcoin addresses are in danger of revealing their complete addresses. They also showed that this data leak is worsened in several cases (i) when nodes are either generate an additional address or restart their SPV clients (ii) when the attacker has access to multiple bloom filters relating to the similar SPV clients. All of the cases require a re-computation process of new Bloom filters, which results in reducing the overall protection of a SPV client. In [13], authors demonstrated that an attacker could exploit scalability and optimization measures to effectively prolong transactions propagation and blocks to particular nodes without causing any hindrance to the network partitioning system. They also demonstrated that this permitted the attacker to mount Denial of Service attacks, significantly enhance its mining advantage within the network, as well as carry out double spending transactions despite the countermeasures applied by Bitcoin. The Bribery attack as explained in [14], where an attacker may use bribery to attain most of the computing possessions for a small amount of time. Authors explained that bribery in the network can be introduced by three models. First is out-of- Band Payment that involves attacker paying directly to the individual that holds the computing resources and further these owners attempt to mine the blocks defined by the attacker. Second is Negative-Fee Mining Pool which involves attackers paying a better returns for creating a pool , and third, In-Band Payment via Forking that involves attackers trying to bribe through Bitcoin itself by forming a fork. By attaining most of the hash power, the adversary can initiate various attacks like DDoS and double spending [15]. In [16], the authors gave a detailed experimental evaluation of Bitcoins DDoS attacks by verifying following: 40 Bitcoin services were attacked by 142 unique DDoS attacks and 7% among these known users became the victims of such attacks. The authors also signified that most of the DDoS attacks are aimed at exchange services and huge mining pools due to the huge amount of money they can earn if successful in attack execution, rather than targeting small mining pools or individuals. In [17] the authors employed a sequence of game-theoretical methods in order to examine the trade-off among two mining pool related strategies. The first strategy is known as construction, where a mining pool significantly invests to enhance its mining capability to improve the probability of winning. The other strategy is destruction where the mining pool initiates a DDoS attack to reduce the success rate of the opposing mining pool.

3.3 Methods to detect DDoS Attack

Different methods have proposed to tackle DDoS attacks which in turns reflect such attacks severe consequences .Due to the failure of detecting DDos attacks quickly and accurately. Still, DDos attacks detection becomes a key research topic in the field of network security.

In [18] authors attempted to detection DDoS attacks proactively by conducting machine learning based on the cluster analysis method. They separated the DDoS attack into various phases, and they found several precursors that were needed for the proactive detection of attacks. Then, they proposed a detection system which could calculate the Euclidean distance between the precursors' entropy values. After which, they identified the connection between them by applying WARD's minimum variance. [19] proposed a novel software defined networking (SDN) based scalable solution called TIDS (transparent intrusion detection system) for the detection of flooding based DDoS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

attacks at the network layer. Shannon entropy measure was employed to detect malicious traffic. In [20] authors attempted to distinguish a HR-DDoS attack based on flow similarity by employing a chaos theory based model. Then, a neural network based system was developed to detect anomalous traffic. A fuzzy logic based method was proposed by Xia et al. in [21] for real-time identification of HR-DDoS and LR- DDoS attacks. Their proposed approach functions in two stages. In the first stage, a statistical analysis of the time series network traffic data was conducted by employing a discrete wavelet transform (DWT). The deviations in the Hurst parameter were found by using a Schwarz information criterion (SIC). Next, in the second stage, attacked packets were identified and then evaluated by counting the number of packets that were dropped. Finally, the authors validated their proposed method through the use of NS2 simulations. In the same context, authors in [22] used traceback approach to defend against DDoS attacks in real-time. They employed mean packet inter-arrival times to construct a fuzzy estimator. Then, the proposed scheme was validated by using DARPA dataset.

In [23], the author suggested such a robust protocol against a DDoS attack which was a Proof of Activity (PoA) protocol. This protocol worked by broadcasting a considerable number of invalid blocks in the network. Within the PoA, a crypt value is assigned to every block header and users keeping the first transaction use such a value. They are known as a stakeholder within the network, and they are thought to be legitimate. In this block, any further arrangement of transactions is carried out if other valid stakeholders are linked with the block. Storing of the crypt values is arbitrary and a large amount of transactions are kept in store only if there are more stake users are linked with the chain. However, more miners are attracted to a particular chain if the length of that chain is increasing, so that reflects an increment of trustworthiness among other peers. Accordingly, an adversary is unable to add a transaction or malicious block because stakeholders are controlling all the nodes in the network. An alternate approach to lessen DDoS attacks is to employ the methods employed in [24], which recommended the constant monitoring and control of the network traffic by employing machine learning methods like clustering and a support vector machine (SVM), or any operator-defined web service like Tor. These methods would recognize which part of the network is acting hostile or is compromised. Hereafter, that particular part could be secluded from the network until it's fixed and debugged. Other likely techniques to secure against DDoS attack contain (i) Alteration of a network configuration in such a manner that prevents malicious requests and packets from unnecessary ports (ii) Apply an additional DDoS protection scheme employing a third-party which would involve monitoring of network and recognizing differences in the pattern.

Indeed, several types of research have been conducted [25]–[28] on effective detection methods with low computational complexity as the network bandwidth increases and the computing speed increases. Accordingly, authors measured the complexity of traffic using entropy and then proposed methods for detecting a denial of service attack based on the analyzed results. Moreover, authors in [29] proposed a method that integrated Blockchain with IoT devices to address DDoS security issues in IoT.

3.4 Defense mechanisms for Blockchain

To prevent the leakage of Bitcoin addresses, authors in [30] introduced a design based on the privacy metric γ-Deniability. Trustzone-based solutions to address some security issues gained more attention and several researchers utilized it for that purpose. A Bitcoin wallet based on Trustzone was proposed in [31]. The authors claimed that this wallet could protect the private key. However, the proposal in [31] is restricted in its applicability on devices with limited hardware because it needs to store all blocks in a local database and it only protects the full nodes wallets. To address this limitation, authors in [32] proposed a lightweight wallet based on ARM's Trustzone technology.

As discussed earlier, the SPV client only needs to connect with the blockchain networks full nodes. During the transaction, SPV client will request from the connected remote node the corresponding transaction information, including the Merkle tree, and a copy of the transaction. Next, the validity of the transaction will be calculated and then verified by SPV client through the information stored in the Merkle tree structure. Nevertheless, the earlier process only reflects the cryptographical protection of SPV client private key. It can be argued that an attacker can still tamper with the SPV client's transaction. Further, if the attacker tampers with a transaction of the target address, the attacker can alter the direction of the money to the wrong wallet. Meanwhile, the certainty of the result of the verification is questionable in case of tampering with the local database which stored the block headers. Thus, it is essential to improve the security

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of SPV client. As the research is still in progress, there is no standard method to protect the verification process of transactions and the local database. This work aims to propose a very simplified DDoS detection method because more accurate detection often requires a high complex computational cost. We acknowledge the fact that the employment of deep neural networks (DNNs) or convolutional neural networks (CNN) or other advanced methods can significantly lead to improving performance. However, as we mentioned above, a trade-off exists. In our method, we investigate the relationship between the incoming packets, which can be effectively used to create rules to detect DDoS attacks in SPV.

4. DESIGN AND METHODOLOGY

4.1 Extract Features Among Packets

Using Association rule learning to detect a DDoS attack can create a challenge, in particular regarding how to define a transaction for analyzing frequent patterns. To overcome that, our method divides connection logs continuously generated into several groups depending on their time stamps, and those in the same group form a transaction. In this approach, a DDoS attack detection operation can be performed by a transaction. Therefore, the smaller the window is, the more frequently the DDoS attack detection operation can be performed. However, a very small window is inefficient for DDoS attack detection, because it is almost impossible to get significantly frequent patterns. The size of the window determines the time when the DDoS attack detection operation can be performed and it affects the detection results usefulness.

In general, a connection log consists of various features, as shown in Table 1.

As a simple and general approach, all connection logs are transformed into a single-target transactional dataset, with no considerations as to the source hosts, the destination hosts, etc. This approach transforms connection logs into a transactional dataset as follows:

1. Connection logs that are generated in the same time window, i.e., those whose time stamps are in the same time window form a transaction.
2. Essential descriptors such as connection logs, service, source host, destination host, and flag are used to define an item and duration etc. are used only to determine the relationship between the transaction and the connection log.
3. For a new connection log, if the values of the four essential features (excluding time stamp) are the same as those of a connection log in a transaction where the new connection log belongs, the new connection log is not considered a new item, but the number of repetitions for the corresponding item is increased by one.
4. Regardless of the values of four essential features in two connection logs being the same, if the logs are generated in different time windows, then each connection log is considered as a separate item in each of the connection logs corresponding transactions.

Table 1 : Features included in a connection log

No	Attribute Name	Label
1	Duration	Basic
2	Protocol type	Basic
3	Service	Basic
4	Source Bytes	Basic
5	Destination Bytes	Basic
6	Flag	Basic
7	Land	Basic
8	Wrong fragment	Basic
9	Urgent	Basic
10	Hot	Content
11	Number of failed logins	Content
12	Logged in	Content
13	Number of Compromised	Content
14	Root shell	Content
15	Attempted	Content
16	Number of Root	Content
17	Number of file creations	Content
18	Number of shells	Content
19	Number of access file	Content
20	Number of outbound commands	Content
21	Is hot Login	Content
22	Is guess Login	Content
23	Count	Traffic
24	Send error rate	Traffic
25	Receive error rate	Traffic
26	Same sever access rate	Traffic
27	Different server access rate	Traffic
28	Server count	Traffic
29	Server send error rate	Traffic
30	Server receive error rate	Traffic
31	Server different host rate	Traffic
32	Destination host count	Host
33	Destination host server count	Host
34	Destination host same server rate	Host
35	Destination host difference server rate	Host
36	Destination host same source port rate	Host
37	Destination host and server host difference rate	Host

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

38	Destination host send error rate	Host
39	Destination host server send error rate	Host
40	Destination host receive error rate	Host
41	Destination host server receive error rate	Host

The DDoS attack is attempted from multiple source hosts or to multiple destination hosts simultaneously. A source-based transformation approach defines as a transaction if only the connection logs are attempted from the same source host. Therefore, the approach may be limited in detecting a general DDoS attack that is simultaneously attempted from multiple source hosts. Likewise, a destination-based approach is also limited in its applicability to detect DDoS attack. However, to monitor a specified host in a network environment that consists of many hosts, a source-based or destination-based transformation can be applied efficiently. In this paper, network connection logs are transformed into a transactional dataset by a simple transformation approach. Then, the frequent mining patterns are performed using the transactional dataset to detect DDoS attack.

Let D denote a network log data set. For a transaction T $\in$ D, when the number of packets contained in T is m, T is represented by$\{p_1, p_2, p_3, \dots, p_m\}$. Also, when the number of items contained in a packet $p_i \in T$ is denoted by n, p_i is represented by $\{d_1, d_2, d_3, \dots, d_n\}$.

Definition 1. (Partial element) Let T be a transaction contained in a connection log set D. For a packet $q \in T$, if $p \subseteq q$, then packet p is called the partial element in transaction T, i.e $p \in T$.

Definition 2. (Partial subset) Let p_s and p_t denote the transaction T_s elements. Then a packet-set $P = \{p_1, p_2, p_3, \dots, p_k\}$ is called the partial subset of the transaction T, where, for all p, and $p_j \in P$, $p_i \subseteq p_s$ and $p_j \subseteq p_t$ $(p_s \neq p_t)$, i.e., $p \subseteq T$.

Definition 1 describes whether a packet p is contained in a transaction T. In other words, if the packet corresponds to any packet in the transaction or its subset, then the packet can become the transaction's partial element. Definition 2 describes the similarity between a packet set and a transaction. In other words, if all packets contained in the packet set are partial subsets, any packet contained in the transaction, the packet set can become the transaction's partial subset. Therefore, the supports of a packet and a packet set can be calculated using Definition 3.

Definition 3. (Support) Let $|E|$ denotes the total number of transactions in E. The supports of a packet p and a packet set P are calculated as Equations (1) and (2), respectively.

$$sup(p) = \frac{|\hat{S}|}{|\hat{D}|},$$

where $\hat{S} = \{p: p \in T \; and \; T \in D\}$ (1)

$$sup(p) = \frac{|\hat{R}|}{|\hat{D}|},$$

where $\hat{S} = \{P: P \subseteq T \; and \; T \in D\}$ (2)

Frequent packet-set mining can be performed as follows:

1. Using Apriori algorithm in [20], [21] , the intra-packet mining is performed. Each frequent packet is transformed by a unique identifier.
2. The connection logs are rewritten by the identifiers of frequent packets generated in the first step.
3. Obtain the frequent packet sets by using the rewritten logs.

4.2 Assumptions

The following assumptions are considered regarding the network model and the proposed technique:

1. SPV devices are considered to be resource constrained with limited memory, power, and processing capabilities.
2. The servers are not constrained in terms of resources.
3. The attacker is on the same local network as the victim.
4. SPV transactions are malleable by third parties, which in turns can enable obscure attacks on SPV verifiers.

4.3 Threat Model

The Bitcoin network has a consensus protocol and distributed nature, thus our suppositions depend on the fact that an adversary is capable to initiate a DDoS attack when more than one attacker initiates the attack at the same time. Actually, malicious miners carry out DDoS attack by gaining access to the competing miners' distributed Botnet and dragging them outside the network in order to successfully enhance the malicious miner's hashrate. Ultimately, the attacker consumes the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

network resources to intervene the access to authentic users.

4.4 DDoS Attack Detection

To detect an anomaly, a normal activity profile is maintained by two elements: a frequent itemset and its item-occurrence vector. The frequent itemset is a profile for representing relationships among network connections in a transaction, whereas the item-occurrence vector is a profile for representing the number of same connections in the transaction. When the number of frequent itemsets is n, let P denotes a set of frequent itemset profiles, i.e., $P = \{p_1, p_2, p_3, \dots, p_n\}$, and each frequent itemset profile is composed of an itemset and an item-occurrence vector. Let V_e denotes the item-occurrence vector of itemset e, represented by Definition 4.

Definition 4. (Item-occurrence vector) When f_a denotes the item occurrence of an item a in an itemset, it means the repetition numbers of the item a, for an itemset $e = \{a_1, a_2, a_3, \dots, a_n\}$, in a resulting set of frequent itemsets, its item-occurrence vector V_e is defined as follows:

$$V_e = \{F_{a_1}, F_{a_2}, F_{a_3}, \dots, F_{a_n}\}. \quad (3)$$

When $C(e)$ denotes the number of a frequent itemset e in a transaction data set D, and D_j is the jth transaction containing e, then $F_{a_1}(a_i \in e)$ denoting the average item occurrence of item a_i is found as:

$$\frac{1}{C(e)} \sum_{j=1, e \subseteq D_j}^{C(e)} f(a_j). \quad (4)$$

Identifying any anomalous behavior of the new transaction can be achieved by comparing a profile set to new online transactions activities. The outcome of this comparison is articulated in terms of both the itemset length and item occurrence differences. The itemset length difference is a measurement that represents the variance existing in between the length of an itemset in a profile and that of the new transaction. Similarly, the item occurrence difference is a measure representing the Euclidean distance between the item-occurrence vector of a profile in the profile set and the new transactions item-occurrence vector. These differences are examined as follows. To detect an anomaly in a new transaction T, a set of frequent itemsets, which are similar to transaction T, are searched for in a profile set. In other words, they are used for the determination of the two variances for transaction T.

For a new transaction $T = \{a_1, a_2, a_3, \dots, a_l\}$, let $MFI\tau$ denotes a set of maximally frequent itemsets for transaction T, and let $\pi e(V_T)$ denotes the vector of item occurrences commonly contained in itemset e and transaction T. Then the Euclidean distance between itemset e's and item-occurrence vectors in the transaction T is represented by

$$d(V_e, \pi e(V_T)) = \sqrt{\sum_{i=1}^{|e|} \left(F_{a_1} - \overline{f_{a_l}}\right)^2}. \quad (5)$$

The itemset length and item occurrence differences are calculated as follows:

$$length\_diff(MPI\tau, T) = 1 - |e \cap T|(e \in MPI\tau, T). \quad (6)$$

$$Ocurence\_diff(MPI\tau, T) = \frac{1}{|MPI\tau|} \sum_{e \in MPI\tau} d(V_e, \pi e(V_T)). \quad (7)$$

Ultimately, for a new transaction T, the total abnormality is given as follows:

$$DDosAttack(MPI\tau, T) = \beta \times length\_diff(MPI\tau, T) + (1 - \beta) Ocurence\_diff(MPI\tau, T). \quad (8)$$

In the above equation, the effects of the two differences, *length_diff* and *occurrence_diff*, can be controlled by setting a proper weight β. In our method, we consider setting different abnormality levels to determine the rate of DDoS behavior in the new transaction T, in which the normal behavior of historical activities relatively defined these abnormality levels. As for defining the status of new objects activities, we classified that based on two different levels (red, green). The red level indicates a warning level and green level means a safe level. Let the denotation of $A(\mu, \nu, \psi)$ present the abnormalities' statistics. While μ, ν and ψ serve as the cumulative number of objects occurrence within a dataset: the square sum and linear sum of their attacks. Accordingly, on the basis of the mean of statistics μ, abnormalities α, and standard deviation σ can be measured as follows:

$$\alpha = \frac{\mu}{\nu} \text{ and } \sigma = \sqrt{\frac{\psi}{\mu} - \alpha^2}. \quad (9)$$

The new object belongs to green or red level, as follows:

- Green Level

$$if \quad 0 \leq DDosAttack(MPI\tau, T) \leq \alpha + \sigma \times \xi. \quad (10)$$

- Red Level

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$if \quad \alpha + \sigma \times \xi < DDosAttack(MPI\tau, T). \quad (11)$$

The new objects strictness of an anomaly is determined by using a detection factor ξ, this detection element is a user-defined parameter, that used for classification of new objects anomaly. A new object is examined strictly with the decrease in detection factor. The ratio of the total number of normal objects and various objects inside the boundaries of the red level indicates the rate of false alarm for a given set of normal objects. Likewise, the total numbers ratio of anomalous objects and the ratio of various objects that are inside the boundaries of the red level gives the rate of anomaly detection for a given set of anomalous objects.

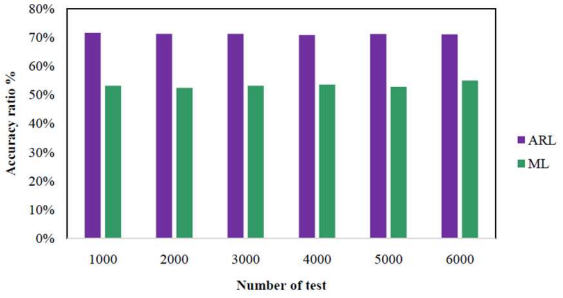
Detection DDoS attack using the ARL

Input: packet traffic flow T, Feature Database FD
// Extract Features from packet traffic
For p in T
FD · ExtractFeature(p)
End
// Detect DDoS Attack
Length_Diff · CalcLengthDiff(FD)
Occurrence_Diff · CalcOccurrenceDiff(FD)
AttackAbnormality · CalcAbnormality (Length_Diff, Occurrence_Diff)
AbnormalityThreshold · Calc AbnormalityThreshold(FD)
If 0< AttackAbnormality < AbnormalityThreshold
Return False;
Else AttackAbnormality > AbnormalityThreshold
Return True

Figure 2. Our proposed Algorithm

5. EXPERIMENTAL RESULT

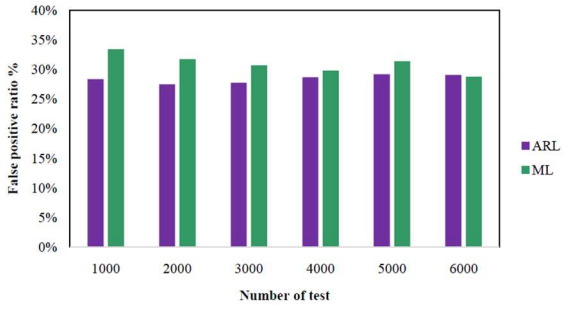
The experiments presented in this paper were performed using the KDD (Knowledge Discovery in databases), NSL-KDD, and GureKDD datasets. KDD NSL-KDD and GureKDD datasets are considered a well-known benchmark datasets in the realm of Intrusion Detection techniques. In this paper, the proposed algorithm based on Association rule learning is simulated in Python environment and compared in compared in terms of the performance results with DDoS attack detection based on machine learning.



	1000	2000	3000	4000	5000	6000
ARL	71.5	71.2	71.2	70.8	71.1	71.0
ML	53.1	52.4	53.1	53.6	52.7	54.9

Figure 3. Detection Accuracy

Figure 3 shows the evaluation accuracy of the DDoS attack detection system due to association rule learning and machine learning respectively. It is noticeable the DDoS attack detection system due to association rule learning is about 30% or above more accurate than machine learning. This shows that the relationship between packets in the DDoS attack is clear and thus association rule learning is very effective in detecting the DDoS attack.

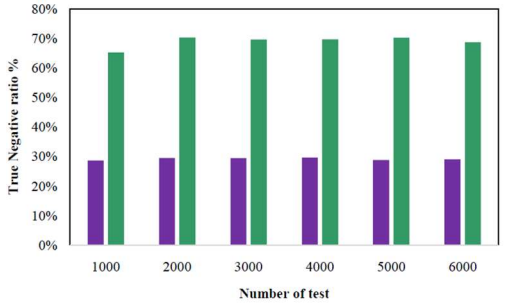


	1000	2000	3000	4000	5000	6000
ARL	3.82	3.85	3.84	3.84	3.84	3.89
ML	10.99	11.08	11.58	11.09	11.15	11.77

Figure 4. False Positive Ratio

Figure 4 presents the evaluation false positive ratio of the DDoS attack detection system due to association rule learning and machine learning respectively. Based on our results, the DDoS attack detection system due to association rule learning begins at about 15% less than machine learning when number of test is one thousand, then it reduces whenever the number of tests increases.

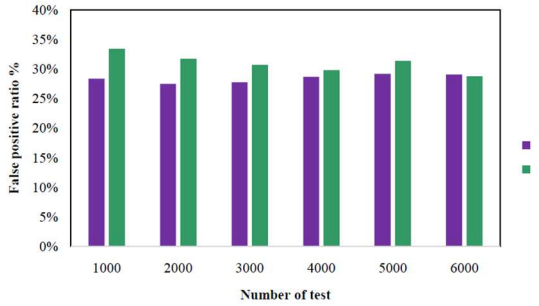
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



	1000	2000	3000	4000	5000	6000
ARL	28.4	27.5	27.7	28.6	29.2	29.1
ML	33.4	31.7	30.7	29.8	31.4	28.7

Figure 5. True Negative Ratio

Figure 5 exhibits the evaluation of true negative ratio of the DDoS attack detection system comes by association rule learning and machine learning as well. Our results show that the DDoS attack detection system based on association rule learning is about 56% or above less than machine learning. True Positive Ratio is less than False Negative Ratio in machine learning. This shows machine learning is based on dataset and Positive data of KDD dataset is better than negative data.



	1000	2000	3000	4000	5000	6000
ARL	28.6	29.5	29.4	29.5	28.8	28.9
ML	65.2	70.3	69.6	69.6	70.2	68.6

Figure 6. Processing Time

The evaluation of the processing time of the DDoS Attack detection system based on association rule learning and machine learning is presented in Figure 6. In this experiment, we can see that the DDoS attack detection system based on association rule learning is about 65% or above less than machine learning. This shows that the association rule learning is simpler than machine learning and indicates that the association rule learning is much faster than the machine learning in DDoS attack detection because the machine learning runs proceeds with dataset-based training.

6. VALIDATION AND DISUSSION

We extended the testing of the proposed method to include different datasets, and we applied our method to NSL-KDD (which is the updated version of KDD dataset) and GureKDD datasets to measure its effectiveness. Based on the result below, we observed many variations once we applied our proposed method to these datasets. Initially, as machine learning algorithm learns patterns from data and association rule learning works on support and confidence values (i.e., the probability of occurrence of same instances). So, in the case of the existence of identical and equal instances in the dataset, which having different categorical values, then the association rule learning can give false results. To reflect the notion above, Figure 7 demonstrates the result of our method once it is applied to a NSL-KDD dataset. NSL-KDD dataset has the same instances of different class values and thus our method shows a high false positive rate compared to machine learning. While, the result we obtained from a GureKDD dataset exhibits almost the same result of a NSL-KDD dataset for Accuracy, and true negative ratios see Figure 8. However, in a GureKDD dataset the normal instances are very high, so the machine learning algorithms are biased towards the imbalance data as the normal instances are higher in numbers, and that why machine learning demonstrated high ratio towards these normal instances. As can be seen from Figure 9, the accuracy rate of our method is not extremely high. However, as the objective of this study is to propose a simple yet effective detection technique for DDoS attacks in SPV based on association rule mining with fewer resources and minimum processing time. Therefore, we evaluated our proposed method on three datasets. Based on the conducted evaluation experiments, our method performed very well when it was compared to machine learning. Although, there were some variations the method achieved excellent results.

7. CONCLUSION

With the rapid development of the Internet, DDoS attack detection is becoming more important than ever. Currently, different techniques including machine learning are used to detect such attacks. However, machine learning is susceptible to errors, the computational amount is very high, and it needs a lot of training data (a big dataset). The association rule mining is the foundation of the data mining models. The level of efficiency and speed of this

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm reflects its quality. The Association rules algorithm is a key data mining algorithm. Our work discussed DDoS attack security implications on SPV client and then the association rule learning application to DDoS attack detection problems. To this end, we proposed a new DDoS attack detection system based on the association rule learning and our method utilized the relationships among received packets in order to detect such attack. In the simulation of our proposed system, the experimental results confirmed that our proposal detected the DDoS Attack accurately and efficiently. In future research, we intend to extract more variables of network traffic in order to analyze it effectively, and then develop an advanced DDoS attacks detection algorithm accordingly.

REFERENCES

[1] https://bitcoin.org/en/developer-guide#simplified-payment- verification-spv

[2] N. T. Courtois, P. Emirdag, and D. A. Nagy, Could bitcoin transactions be 100x faster? in 2014 11th International Conference on Security and Cryptography (SECRYPT), Aug 2014, pp. 16.

[3] A. Gervais, S. Capkun, G. O. Karame, and D. Gruber, On the privacy provisions of Bloom filters in lightweight bitcoin clients, in Proceedings of the 30th Annual Computer Security Applications Conference on - ACSAC 14, 2014.

[4] Z. Ihsan, M. Y. Idris, K. Hussain, D. Stiawan, and K. Mahmood Awan, Protocol Share Based Traffic Rate Analysis (PSBTRA) for UDP Bandwidth Attack, in Informatics Engineering and Information Science, Springer Berlin Heidelberg, 2011, pp. 275289.

[5] D. C. Wyld, M. Wozniak, N. Chaki, N. Meghanathan, and D. Nagamalai, Eds., Advances in Network Security and Applications. Springer Berlin Heidelberg, 2011. https://doi.org/10.1007/978-3-642-22540-6

[6] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj, A Survey on Security and Privacy Issues of Bitcoin, IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 34163452, 2018.https://doi.org/10.1109/comst.2018.2842460

[7] J. A. Kroll, I. C. Davey, and E. W. Felten, The Economics of Bitcoin Mining or, Bitcoin in the Presence of Adversaries, Workshop on the Economics of Information Security, 2013

[8] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, Is bitcoin a decentralized currency? IEEE Security Privacy, vol. 12, no. 3, pp. 5460, May 2014.

[9] L. Hanguang and N. Yu, Intrusion Detection Technology Research Based on Apriori Algorithm, Physics Procedia, vol. 24, pp. 16151620, 2012.

[10] J. Mirkovic and P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, p. 39, Apr. 2004.

[11] T. M. Gil and M. Poletto. MULTOPS: a data-structure for bandwidth attack detection. In Proceedings of 10th Usenix Security Symposium, August 2001.

[12] V. Sekar, N. Duffield, O. Spatscheck, J. van der Merwe, and H. Zhang. LADS: Large-scale automated DDoS detection system. In USENIX Technical Conference, June 2006.

[13] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun. Tampering with the delivery of blocks and transactions in bitcoin. Cryptology ePrint Archive, Report 2015/578, 2015. http://eprint.iacr.org/.

[14] B . J., Why buy when you can rent? Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science, vol 9604. Springer, Berlin, Heidelberg, 2016.

[15] A. F. Neil Gandal, Tyler Moore and J. Hamrick, The impact of ddos and other security shocks on bitcoin currency exchanges: Evidence from mt. gox, The 15th Annual Workshop on the Economics of Information Security, vol. abs/1411.7099, June 13- 14, 2016.

[16] M. Vasek, M. Thornton, and T. Moore, Empirical analysis of denial of-service attacks in the bitcoin ecosystem, in Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Springer Berlin Heidelberg, 2014, pp. 5771.

[17] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, Game-theoretic analysis of ddos attacks against bitcoin mining pools, in Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014,. Springer Berlin Heidelberg, 2014, pp. 7286.

[18] K. Lee, J. Kim, K. H. Kwon, Y. Han, and S. Kim, Ddos attack detection method using cluster analysis, Expert Systems with Applications, vol. 34, no. 3, pp. 16591665, 2008.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[19] O. Joldzic, Z. Djuric, and P. Vuletic, A transparent and scalable anomaly-based dos detection method, Computer Networks, vol. 104, pp. 2742, 2016.

[20] A. Chonka, J. Singh, and W. Zhou, Chaos theory based detection against network mimicking ddos attacks ,IEEE Communications Letters, vol. 13, no. 9, 2009.

[21] Z. Xia, S. Lu, J. Li, and J. Tang, Enhancing ddos flood attack detection via intelligent fuzzy logic, Informatica, vol. 34, no. 4, 2010.

[22] S. N. Shiaeles, V. Katos, A. S. Karakos, and B. K. Papadopoulos, Real time ddos detection using fuzzy estimators computers & security, vol. 31, no. 6, pp. 782790, 2012.

[23] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, Proof of activity: Extending bitcoins proof of work via proof of stake [extended abstract]y, SIGMETRICS Perform. Eval. Rev., vol. 42, no. 3, pp. 3437, Dec. 2014.

[24] P. Camelo, J. Moura, and L. Krippahl, CONDENSER: A graph- based approach for detecting botnets, CoRR, vol. abs/1410.8747, 2014.

[25] A. Koay, A. Chen, I. Welch, and W. K. G. Seah, A new multi classifier system using entropy-based features in DDoS attack detection, in Proc. Int. Conf. Inf. Netw. (ICOIN), Jan. 2018, pp. 162167

[26] A. T. Lawniczak, H. Wu, and B. Di Stefano, Entropy Based Detection of DDoS Attacks in Packet Switching Network Models, in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, 2009, pp. 18101822.

[27] K. Li, W. Zhou, S. Yu, and B. Dai, Effective DDoS Attacks Detection Using Generalized Entropy Metric, in Algorithms and Architectures for Parallel Processing, Springer Berlin Heidelberg, 2009, pp. 266280.

[28] S. Yu and W. Zhou, Entropy-Based Collaborative Detection of DDOS Attacks on Community Networks, in 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), 2008.

[29] U. Javaid, A. K. Siang, M. N. Aman, and B. Sikdar, Mitigating loT Device based DDoS Attacks using Blockchain, in Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems - CryBlock18, 2018.

[30] K. Kanemura, K. Toyoda, and T. Ohtsuki, Design of privacy- preserving mobile bitcoin client based on -deniability enabled bloom filter, in Proc. 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC), Oct. 2017, pp. 16

[31] M. Gentilal, P. Martins, and L. Sousa, TrustZone-backed bitcoin wallet, in Proc. 4th Workshop Cryptogr. Secur. Comput. Syst., 2017, pp. 2528.

[32] W. Dai, J. Deng, Q. Wang, C. Cui, D. Zou, and H. Jin, SBLWT: A Secure Blockchain Lightweight Wallet Based on Trustzone, IEEE Access, vol. 6, pp. 4063840648, 2018.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

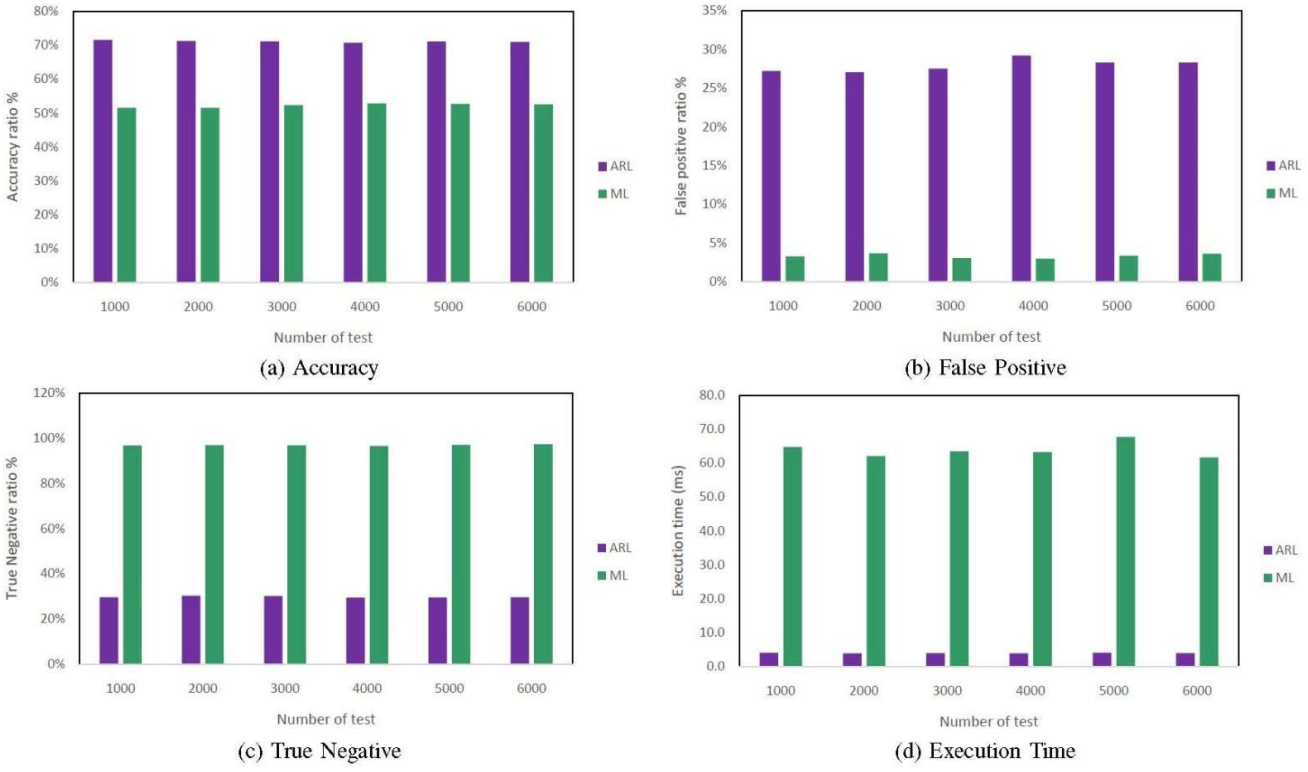


Figure 7. Results obtained from NSL-KDD dataset

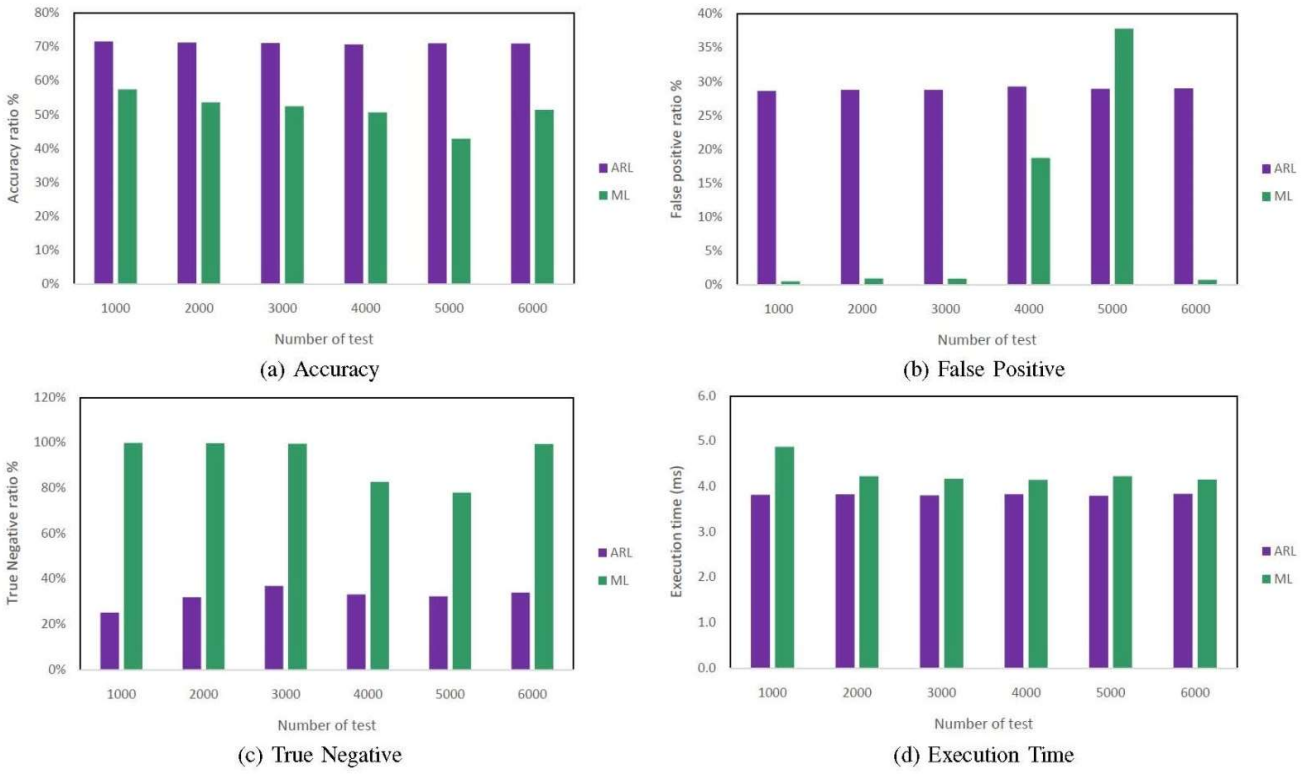
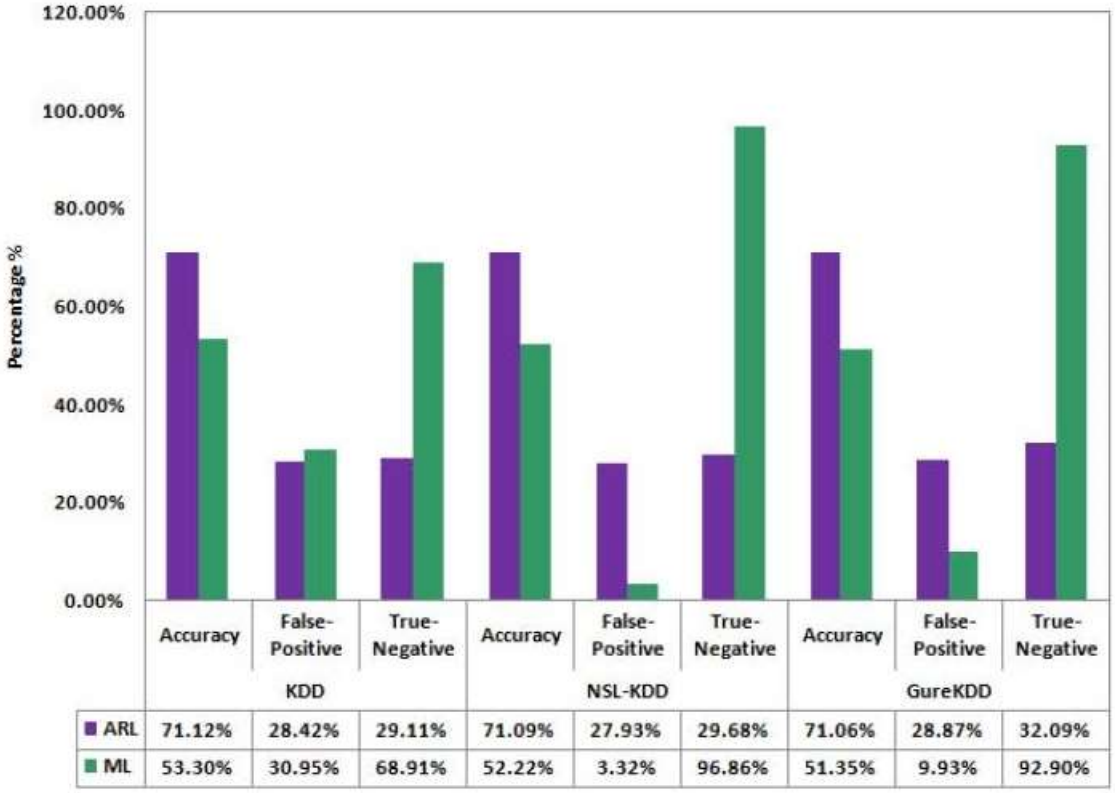
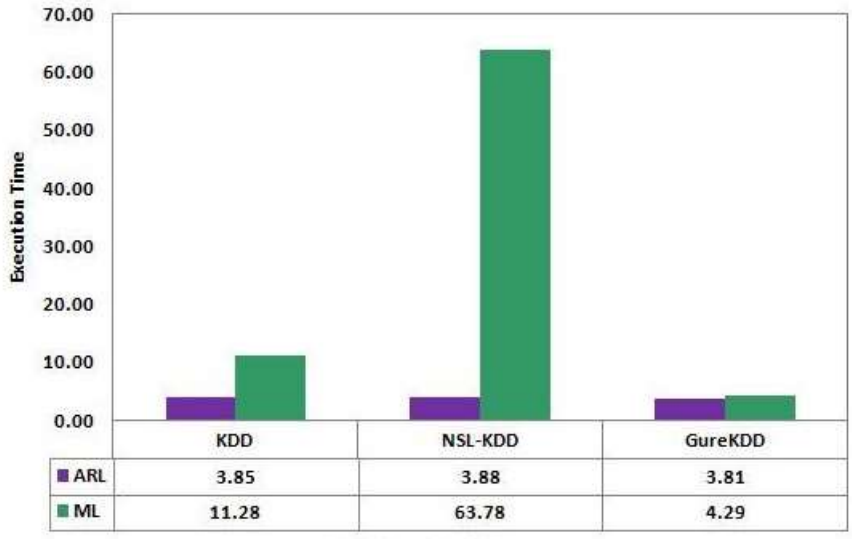


Figure 8. Results obtained from GrueKDD Dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a) Accuracy, False Positive and True Negative



(b) Execution Time

Figure 9. Comparison between ARL and ML for KDD, NSL-KDD and Grue-KDD datasets