Toward an Ontological Cyberattack Framework to Secure Smart Cities with Machine Learning Support

Ola Malkawi[1], Nadim Obaid[2]
Computer Science Department
University of Jordan
Amman, Jordan

Wesam Almobaideen[3]
Electrical Engineering and Computing Sciences
Rochester Institute of Technology
Dubai, UAE

***Abstract*—With the emergence and the movement toward the Internet of Things (IoT), one of the most significant applications that have gained a great deal of concern is smart cities. In smart cities, IoT is leveraged to manage life and services within a minimal, or even no, human intervention. IoT paradigm has created opportunities for a wide variety of cyberattacks to threaten systems and users. Many challenges have been faced in order to encounter IoT cyberattacks, such as the diversity of attacks and the frequent appearance of new attacks. This raises the need for a general and uniform representation of cyberattacks. Ontology proposed in this paper can be used to develop a generalized framework, and to provide a comprehensive study of potential cyberattacks in a smart city system. Ontology can serve in building this intended general framework by developing a description and a knowledge base for cyberattacks as a set of concepts and relation between them. In this article we have proposed an ontology to describe cyberattacks, we have identified the benefits of such ontology, and discussed a case study to show how we can we utilize the proposed ontology to implement a simple intrusion detection system with the assistance of Machine Learning (ML). The ontology is implemented using protégé ontology editor and framework, WEKA is utilized as well to construct the inference rules of the proposed ontology. Results show that intrusion detection system developed using the ontology has shown a good performance in revealing the occurrence of different cyber-attacks, accuracy has reached 97% in detecting cyber-attacks in a smart city system.**

***Keywords*—Cyberattack; Internet of Things (IoT); ontology; machine learning; intrusion detection system**

I. Introduction

The Internet of Things (IoT), defines the large number of devices that can be connected to the internet and perform different types of work. Different devices and sensors can provide our life with digital intelligence, which can serve peoples' needs with minimum or zero human intervention [1]. IoT devices are connected to each other as well as to the internet via a computer network. It is worth mentioning that wireless networks dominate the connectivity between IoT devices which may increases the opportunity for more potential attacks be launched [2].

Features of IoT facilitate the automation of a wide variety of applications and systems, such as health care, homes, traffic lights, and electricity grids to get a smart healthcare, smart home, smart traffic, and smart grids, respectively, as well as many other services and applications. Hence, the majority of recommended services of a city has been automated which creates the concept of a smart city [3]. The concept of a smart city has gained a large concern from governments and business agents as it plays a vital role in the progress and development of the new understanding of civilization in modern countries.

However, security is one of the most prominent challenges when we deal with smart life aspects such as smart cities. This is because developing intrusion detection or prevention systems to secure smart systems is not an easy task, especially with the continuous emergence of new attacks. This raises the need for a uniformed understanding of cyberattacks [4, 5]. The main goal of this uniformed understanding is to develop suitable protection tools for a certain category of attacks, which, at the same time, can protect the system against potential attacks which could appear in the future. Developing an ontology for cyberattacks can provide this formal and uniformed representation, which may provide a general base for a certain category of attacks based on predefined criteria.

Developing an ontology for cyberattacks can also help to understand needed characteristics of a certain system before selecting the protection method. Strictly speaking, different organizations may be interested in different security concepts based on the organizational type and function. A newspaper information system, for example, may consider integrity and authentication of the published news to be of great concern whereas, a healthcare system highly considers confidentiality and privacy of the exchanged patients' information to be essential [6]. By defining the security needs, potential cyberattacks and their impact on the system can be defined and characterized which could play a major role in developing and selecting the suitable protection system.

The contribution of this paper can be summarized with the following points:

1) Proposing an ontology for cyberattacks of smart cities in the context of IoT.

2) Defining the benefits of developing an ontology for cyberattacks from deferent perspectives.

3) Presenting a formal representation and implementation of the proposed ontology using Description Logic and protégé software to conduct reasoning.

4) Using ML as a tool to define the inference rules for the proposed ontology.

5) Using the proposed ontology to pick up the features needed to apply ML.

6) Integrating ML model with the implemented ontology to develop a simple knowledge base for certain attacks.

The rest of this paper is organized as follows. In Section II we present the most important related work. Section III illustrate the followed methodology in conducting this research. A case study is discussed in Section IV which is related to using machine learning to Secure Smart City which also include the conducted experiment and discussed results. We conclude the paper with Section V.

II. Related Work

In cyber-security research area, developing ontologies is not new. A number of approaches have investigated ontology to develop or design a security framework [4]. In this section, we summarize some state-of-the-art research works focusing on the development of ontologies in the context of security and privacy.

Proposed ontologies of cyber-security can be categorized as follows: (1) ontologies that considered information security, (2) ontologies that considered security in IoT, and (3) ontologies that considered security in smart city. Fig. 1 presents the hierarchy of ontologies of cyber-security in state-of-the-art.

In category (3), authors of [7] and [8] have proposed ECA and OBPP ontologies, respectively. Both ontologies have considered only privacy issues with the use of cloud computing in smart cities. The other security requirements such as availability, integrity and confidentiality are not considered. In [9], authors have focused on setting guidelines to develop a secure and safe smart city system based on using ontologies. Nevertheless, no ontology is provided in the research paper. In [10] authors have proposed an ontology for cyberattacks in a smart city system. The ontology concentrated on securing smart city applications rather than cyber-attacks. Moreover, authors provided a number of use cases with mapping each use case with the proposed ontology with no inference rules to get a benefit from the proposed ontology.

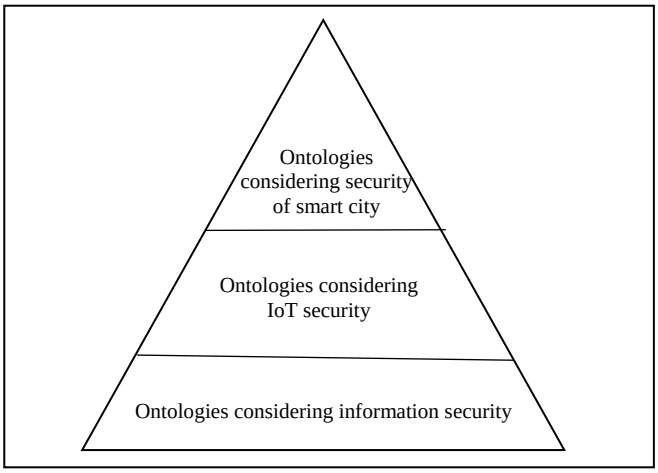


Fig. 1. Hierarchy of Ontologies of Cyber-Security in State-of-the-Art.

In category (2), authors in [11] proposed an ontology to secure an IoT system, the ontology is designed based on analyzing IoT system's vulnerabilities. This work has focused on power-IoT cloud systems which are limited to power issues. Inference rules are set based on vulnerability analysis.

In [12], an ontological analysis has been proposed to enhance security services in IoT systems. The ontology has focused on the system vulnerabilities, potential threats and security needs, it does not elaborate cyber-attacks and their specifications and symptoms.

In [4], the ontology's objective is to create a unified representation for heterogeneous data generated by IoT devices, this work has focused on cyber-attacks and their properties, however, it considers only the general aspects of cyber-attacks. In our proposed ontology, we have concentrated on the cyber-attack properties and their detailed impact on the performance of the network.

In [13] an ontology is proposed to be utilized for higher security improvement in terms of the heterogeneity in the layered cloud platform, the ontological design has focused on the IoT environment and assets such as security devices rather than attack details.

Finally, in category (1), we present a number of these works to illustrate its main structure and focus. In [14] and [15], two cybersecurity ontologies are built by expanding existing ontologies. The main focus of the proposed ontology in [14] is the environment rather than the system itself, and proposed ontology is very general and the main goal is to improve cybersecurity awareness to make suitable decision by providing safe operations rather than detection malicious behavior. In [15], the cybersecurity ontology has concentrated on finding qualification metrics to assess how much a certain system is secure.

In [16] and [17], the main concern of the proposed ontologies is to create an organized schema for cyber information. The goal of using these ontologies is data analysis. While authors of [16] have focused on virus threats and IP and DNS problems, the main focus in [17] is system assets and how to protect these assets.

In [18], authors concentrated in the proposed ontology on data sources and users and potential threats based on these two elements. Rules of ontology are set based on cybersecurity standards and concepts.

Authors of [19] proposed Unified Cybersecurity Ontology (UCO), the main concern of UCO is to identify sources of attacks. Inference is related to derive possible sources of attacks. However, the proposed ontology cannot help to improve security when mapped to IoT. In [20], a cybersecurity ontology is proposed to find guidelines for security measures to protect critical infrastructure, the main focus in this ontology is system assets. Both ontologies in [19] and [20] are customized for specific platforms and to protect simple software. Moreover, most of such ontologies are not suitable on IoT environment according to its specifications and limitation.

Most existing research works are either environment-centric, assets centric, or threats-centric which is opposed to our ontology which is attack-centric scheme. This motivates us to develop Cyber Attack Ontology (CAO) for IoT-based smart city. Moreover, the process of setting inference rules in the previous research works depends on either the analysis of the systems' vulnerabilities and threats, or it depends on user-defined rules. In our approach, we have proposed the use of ML to set inference rules based on the superiority of artificial intelligence and ML in the security during the last years [21][22][23].

III. Methodology

In this section, we present and discuss the proposed ontology, we then show the steps of utilizing the ontology to identify cyber-attacks. In order to make accurate reasoning to identify cyber-attacks, precise rules must be defined. However, and because there is no scientific base to set the inference rules for our ontology, like those in other scientific fields, we will use ML to set up the rules, definitions of attacks and other concepts included in the proposed ontology, a detailed discussion is represented in the next sections.

A. Cyber-attack Ontology (CAO)

In this section we discuss the proposed ontology which is shown in Appendix A. The ontology graph in Appendix A was depicted using protégé OWL [24]. This plugin represents a visual notation for OWL ontologies and a graphical view for the ontology's classes and relations which are joined together to shape a directed graph layout for the ontology. The proposed ontology includes three main entities, which are adversary, system and cyberattack. The adversary is the person who designs and develops a malware or acts in a malicious behavior to launch a cyberattack. There are numerous types of attackers and there are multiple goals for an attacker to launch an attack. A system is described by a number of concepts such as its functionality, components, security needs, performance aspects and system's vulnerabilities. Fig. 2 shows the classes of the proposed ontology after we have implemented this ontology using protégé OWL. Adversary, Cyber-attack and IoT System present the main classes. They are subclasses of the general built-in class Thing initiated by protégé.

Fig. 2. Classes of Cyber-Attack Ontology in Protégé Owl.

A cyber-attack which is the major entity and the subject of this research connects an adversary with the target system of the attack from multiple perspectives. Examples are that the cyber-attack violates one or more of the security principles of the system, the cyber-attack may disrupt the system's functionality, the cyber-attack may degrade the system's performance (and may not) by exploiting the system's vulnerabilities. The cyber-attack also is characterized by the network type which is corrupted with it, the channel through which it is being launched, the protocol or protocols that have vulnerabilities through which the attack could be launched, the network layer for that attack, and the countermeasure adopted against the attack.

Concepts of CAO are shown in Appendix A along with other considered concepts and features. This ontology could be thought of as a base for a wider ontology, more concepts and relations could be added and investigated.

From the discussed ontology, we notice that we can get the following suggested benefits:

1) Ontology helps to develop a taxonomy for cyber-attacks, any entity in the discussed ontology can be picked up as a classification criterion, for example by the selection of channel entity, cyber-attacks could be classified as software-based attacks, hardware-based attacks and network attacks. If we select access level entity, cyber-attacks could be classified as either passive or active attacks, and so on.

2) Ontology provides a formal and unified description and understanding for cyber-attacks and security needs for IoT environment, especially with the heterogenous nature of IoT devices and IoT systems.

3) Ontology can be utilized to define security needs and the protection method for a certain system, for example, for a smart tourism system, we care about the integrity and availability rather than confidentiality, as tourism information tend to be public [25]. By defining our security needs, we can utilize the ontology by navigating ontology structure from one concept to another to define all related needs, costs and components in order to develop a suitable protection system.

4) From an ontology, we can extract features to apply ML which has got a considerable attention during the last decades for a wide variety of applications and especially in the security field.

5) Ontology enables automated reasoning about cyber-attacks, if we build a strong ontology, reasoning could be used to develop semantic graph database for cyber-attacks and all related characteristics such as effect, cost, and suitable countermeasures.

6) Ontology is easy to be extended or changed because we can add concepts and relationships. So, the proposed ontology can evolve with the emergence of new concepts or when discovering any wrong facts with no impact on the existing systems.

In this work, the intended use of the proposed ontology is to navigate its concepts starting from a pre-specified system. With the target is to construct a model to identify cyber-

attacks of that system using ML. We have conducted a case study to build the model, the case study details along with the model designed are discussed in the next sections.

B. CAO Implementation

In this section, we present the details of Cyber Attack Ontology (CAO) including main classes, sub-classes, properties and other elements of ontology. We have implemented the proposed CAO using protégé tool. Fig. 2 presents the main classes of CAO. We have implemented main classes and sub-classes. Then, we have defined properties of our ontology, which are divided into two types: object properties and data properties. Fig. 3 and Fig. 4 present object properties and data properties, respectively. Fig. 3 shows also a representation of the hierarchy of object property "exploits" which has the domain "cyber-attack" and range "vulnerability".

Data properties relates a class to an attribute data such as "integer"," float", "string", ...etc. Fig. 4 presents a number of data properties in CAO such as traffic_sent which indicates the total number of packets sent through the network during a period of time. Traffic_sent has the domain traffic and the range integer.

Rules of CAO are not defined at this phase as they will be derived at the last step where ML will be utilized to set CAO rules.

Finally, we have shown individuals of the ontology, these individuals represent fundamental components of the ontology and they include concrete instances of ontology. In CAO, for example, sinkhole attack is an instance of a cyber-attack class. Sinkhole attack can be described by multiple properties. In this work, a cyber-attack is described by the network performance resulted by the attack occurrence. The performance is presented by network traffic measurements. For example, sinkhole attack can be described with a certain PDR, delay, overhead, etc. Rules to organize and control these properties will be set using ML at section 4.5.

Fig. 5 presents data properties assertions of class traffic. Network traffic is extracted for a period of time and analyzed, performance metrics values such as packet delivery ratio, number of sent packets, and power consumption are calculated, and the resulted values of these metrics are added as assertions for data properties shown in Fig. 4.

In CAO we have defined each cyber-attacks with a certain traffic specification, or in other words, a traffic with particular values of the aforementioned performance metrics is considered as an attack.

C. Description Logic for CAO

Description logic (DL) is one of the formal languages used for knowledge representation. DL is used in artificial intelligence applications to conduct reasoning from related concepts. In CAO, reasoning is needed to determine if there is an attack or not. In this section we present a number of concepts and relations of our proposed ontology, CAO, using DL. Fig. 6 presents DL for CAO.

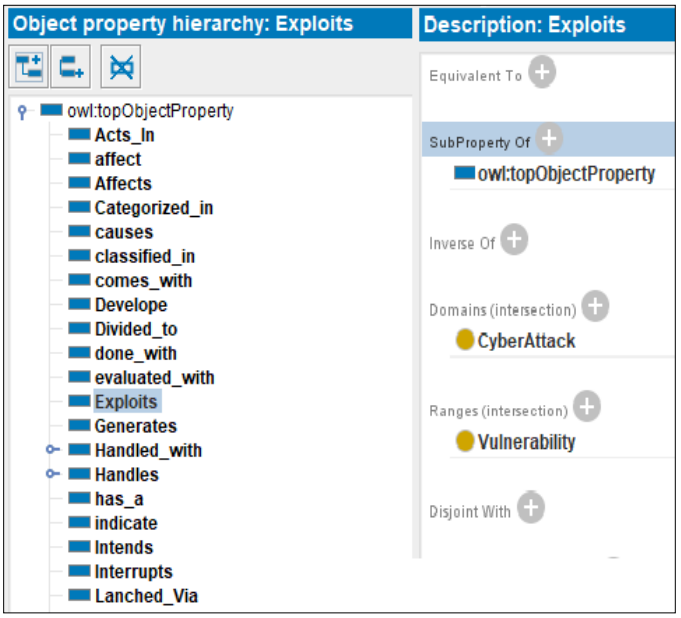


Fig. 3. Object Properties of Cyber-Attack Ontology and Object Hierarchy "Exploits".

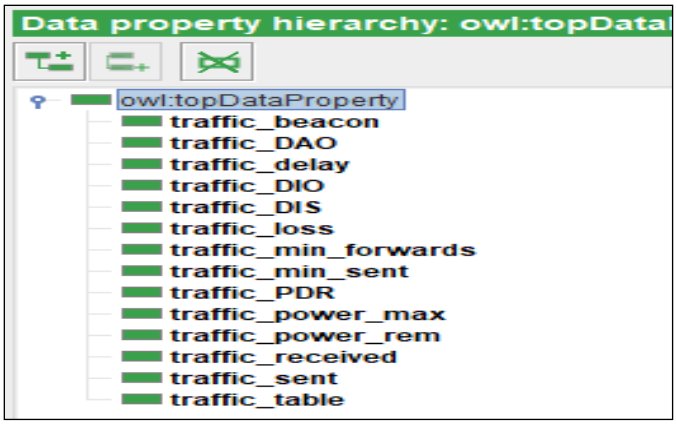


Fig. 4. Data Properties of Cyber-Attack Ontology.

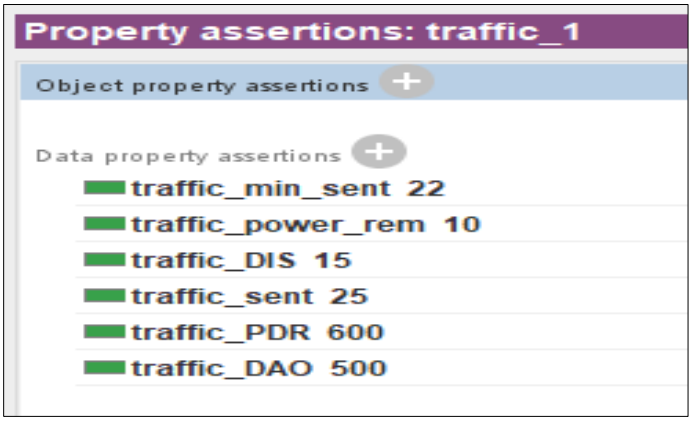


Fig. 5. Data Property Assertions of a Traffic Instance (Traffic_1).

Cyber-attack ⊑ ∃ exploits. Vulnerability	Adversary ⊑ ∃ Develops. Malware
Cyber-attack ⊑ ∃ Interrupts. Functionality	Adversary ⊑ ∃ Acts_In. Malicious_Behaviour
Cyber-attack ⊑ ∃ Categorized in. Category	Adversary ⊑ ∃ Classified_in. Type
Cyber-attack ⊑ ∃ Comes_with. Benefit_for_attacker	Adversary ⊑ ∃ Intends.Benefit_for_attacker
Cyber-attack ⊑ ∃ Done_with. Access_Level	IoT_System ⊑ ∃ Must_Fullfill. Functionality
Cyber-attack ⊑ ∃ Degrades. Performance	IoT_System ⊑ ∃ Maintains. Security_Essential
Cyber-attack ⊑ ∃ Targets. Component	IoT_System ⊑ ∃ Has_a. Vulnerability
Cyber-attack ⊑ ∃ Handled_with. Countermeasure	IoT_System ⊑ ∃ Evaluated_by. Performance
Cyber-attack ⊑ ∃ Affects. Traffic	Performance ⊑ ∃ Indicates. Cyber-attack
Cyber-attack ⊑ ∃ Launched_Via. Network_protocol	Performance ⊑ ∃ Measured_by. Evaluation_metric
Cyber-attack ⊑ ∃ Launched_Through. Network_layer	Performance ⊑ ∃ Measured_Through. Traffic
Cyber-attack ⊑ ∃ Launched_Using. Channel	Network ⊑ ∃ Generates. Traffic
Malware ⊑ ∃ Launch. Cyber-attack	Countermeasure ⊑ ∃ Preserves.Security_Essential

Fig. 6. Description Logic Representing CAO.

D. Ontology to Secure an IoT System

The main utility for cyber-attacks ontology in this work is to design and implement an intrusion detection system to secure a smart city. The first phase of designing any protection mechanism is to study the system to be secured and to highlight the main issues, risks and security needs. We will discuss how to go through these processes using the proposed CAO.

Before proceeding with designing and implementing a security mechanism for a certain system, we have to analyze that system and understand its needs precisely. As we mentioned before, security needs vary from one system to another depending on system's structures and goals. For instance, the most important security needs for a banking system are privacy, confidentiality and integrity, however, a newspaper or tourism systems consider integrity and availability as the most significant requirements to be maintained rather than privacy or confidentiality [25].

Thus, the first step is to determine the most important security needs for the targeted system. Then other elements can be defined in the same way, we can use and track the ontology for a specific system to determine the following proposed elements:

1) The most important security essentials needed to be maintained for the concerned system.

2) The potential domain of a cyber-attack which must be specified carefully. For instance, if computer networks are a potential domain. We have to specify network layer, network protocol and network type whether wired or wireless.

3) Vulnerabilities of the system.

4) The category of the most potential or damaging cyber-attacks that we need to handle.

5) System's components that are exposed to cyber-attacks must be specified, this component could be a computer network, file server, mail server, or any other component.

6) The functionality of each component specified in point number 5 which can be interrupted by a cyber-attack must be also determined.

7) If the performance of the system is of a considerable importance, the performance metrics should be identified.

8) If specifications of the adversary are of a significant importance, then attacker's type and benefit of a cyber-attack are determined, the tactic of the adversary whether designing a malware or acting in a certain behavior are also specified.

9) Finally, and based on the previous points, we can decide what is the most appropriate countermeasure(s) to be designed to secure the concerned system. These may include intrusion detection system, intrusion prevention system or a mitigation method.

IV. Case Study: Ontology to Secure Smart City Using Machine Learning

Wireless sensor networks (WSNs) play a significant role in smart city services. This is due to the fact that sensors comprise most of monitoring and automation systems which are the backbone of smart cities [26]. Actually, sensor networks are fundamental component in smart grids, home automation systems, traffic systems, health care applications, power system monitoring and many other smart city services. However, sensors are limited and constrained devices in terms of capabilities and resources such as power, processing and storage. This makes it harder to secure a sensor network using the common and traditional security means such as cryptography. More efficient and low-cost methods need to be investigated to secure sensor networks. ML is considered as one of the most suitable candidates in this context. ML can be employed to design an intrusion detection method by training the system to discriminate between normal traffic and traffic of a network under attack [27] [28].

To apply ML, we must have a suitable dataset, in the following subsections, we present the steps by which we use the proposed ontology to create a sample dataset with suitable features. We have opted to extract the dataset from the original network traffic, this traffic is generated from normal operations of the network. Thus, we do not add any additional packets or control message to get features of the dataset, which considers limited power of the sensors, so, we do not add any additional load onto the sensor nodes.

A. Ontology Navigation to Identify Main Entities

We have used CAO to make a decision about the features of the dataset, we have explored the ontology and specified the most significant entities. Phases of exploring the ontology that have been discussed in the previous section are applied to the selected case study. These phases are summarized in the following points:

1) The major security principle to be maintained is the availability of the sensor network.

2) Because network is the most potential domain to launch most cyber-attacks of WSNs. We have to specify network related elements including layer and network protocol through which an attack is launched. We have selected RPL routing protocol for our case study which belongs to network layer. Motives behind the selection of RPL protocol can be summarized in its convenience with the requirements of smart and its properties of self-configuration, self-restoration, and the ability to meet power consumption constraints.

3) The main vulnerabilities of WSNs in the light of sensors limitations and RPL protocol can be summarized as: lack of infrastructure, lightweight protocols which are not supported with security mechanisms, constraint devices where cryptography is not efficient, limited physical security, dynamic topologies, unreliable links, multi-hop transmission paradigm which acts as a helping factor to transmit and spread malicious messages, and finally, the distributed problem handling, which means that a malicious node can select the action of not solving a problem which can simply cause the entire network to be malfunctioning, such as local repair attack[29].

4) The most well-known attacks for sensor network regarding availability are specified. These attacks include sinkhole attack, wormhole attack, sybil attack, rank attack, flooding attack, copycat attack. These types of attacks are categorized within denial of service (DoS) attacks.

5) The system component to be secured is a sensor network which is considered as a vital part of smart city system.

6) The functionality of the sensor network which could be interrupted is the process of data transmission from all sensors to a central point. The role of each sensor is to monitor and record certain data in some location and send the sensed data to a central point (sink). We can explain the functionality as that the sensed data must be received by the sink within a certain period of time and entailed with the original sender.

7) The performance of sensor network is of a significant importance. For example, it is recommended that the transmitted packets are received within an acceptable period of time, which is referenced as delay. Moreover, all packets, or at least a satisfying portion, must be received by the intended receiver, which is commonly defined as packet delivery ratio. Both delay and packet delivery ratio can be considered as performance metrics for sensor networks. The next section describes how we have analyzed selected attacks to address the main performance metrics affected by each attack.

8) Characteristics of the adversary such as its type, benefit, or tactic are not customized in this work, the involved attacks include all these variations.

9) The countermeasure to be designed and implemented is an intrusion detection system that employs ML. The features of the dataset will be selected based on the performance of the sensor network, performance indicators or metrics will be extracted from the network traffic.

B. Impact of Studied Cyber-attacks on Performance

The targeted security principle of this work is availability, which means that the service provided by the WSN must be available when needed and with satisfying quality. Thus, we aim to detect attacks that cause the WSN to become un-available or degrade its performance. Therefore, performance must be defined using suitable performance metrics. We have explored the ontology for the selected cyber-attacks to address performance metrics that could be affected for each attack. Performance metrics defined in this section will be considered as the features of the generated dataset:

Rank Attack: this attack interrupts the balance of routing paths distribution in a WSN uses RPL. The first result is the high congestion and interference within the attacker's zone, which causes packet loss, increases end to end delay, and decreases throughput and packet delivery ratio. Moreover, the un-optimal paths will be created and used which may increase power consumption. Furthermore, and as the rank attacker becomes the preferred router for many nodes, the length of routing table will be increased unusually in the attacker and its neighbor nodes. Therefore, the length of routing table could be considered as a feature in the generated dataset.

Sinkhole attack: this attack is similar to rank attack in its effect, however, the attacker can attract more nodes than rank attacker because it claims that it has the minimum rank. Nonetheless, it is easier to be detected. Therefore, the performance metrics affected are packet loss, end to end delay, throughput, packet delivery ratio, and power consumption.

Flooding attack: this attack is executed by broadcasting a large amount of control messages. Hence, we expect a noticeable increase in control packets. Furthermore, the generated traffic will affect end to end delay as a result of increasing interference. Nodes will spend more time in replying to the deceptive traffic which affects throughput and power consumption, Moreover, the resulted interference can increase packet loss which affects delivery ratio. Finally, the deceptive control packets may force nodes to set protocol related metrics, such as the frequency of sending hello packets, unusually.

Copycat attack: as a result of receiving old versions of control packets during this attack, nodes may adjust protocol metrics illogically, such as time interval between hello packets and routing information. Moreover, frequent and unnecessary control packet transmission can exhaust power resources.

Sybil attack: by receiving packets with fake sender's information, nodes build a faulty routing table, For example a node could be inserted in the routing table as a neighbor while it is actually far away or even it does not exist, or a node may be recorded as an optimal router while it presents a high-cost router. The result of such cases is a high packet loss and a higher cost routing paths which leads to a lower packet delivery ratio and higher power consumption and end to end delay.

Wormhole attack: the effect of wormhole attack varies based on the target of the adversary from the established tunnel. For example, if the target is to drop packets this will affect packet delivery ratio. In general, we expect longer and un-optimized paths to be created. So, the foreseeable impact will be relevant to power consumption, end to end delay and packet delivery ratio. Table I, lists the considered attacks as well as the performance metrics affected by each attack.

So far, we have specified targeted security principle that we want to preserve, which is availability, we have then defined attack category to be considered which is (DoS). Based on this category, we have specified six types of attacks from this category for our case study. Then, more navigation through the ontology has ultimately led us to the fact that these attacks affect the performance of the network, and the concerned performance metrics were defined. This has been illustrated in Table I. To set up the rules of the ontology, the most significant step is to determine what is the threshold for each metric that can be considered to decide if there is an attack or not. For instance, what is the value of PDR that represents the lower limit of accepted PDR, and below it we decide that there is an attack? Since it is not easy to guess or evaluate these values, so we will utilize ML to set threshold values and to set rules of the ontology, and developing a model for attack detection in WSNs.

C. Dataset and Cyber Attacks

To set the rules of the proposed ontology, we have utilized ML. This is because there is no scientific rule or base for performance metrics by which we can detect cyber-attack occurrence based on performance. Thus, we applied supervised learning on labeled dataset that include the investigated attacks. The objective of using ML is not to construct the intrusion detection system (IDS), it comes as a complementary step to denote the importance of ontology in defining system needs and potential attacks, which facilitate the process of developing IDSs as well as other protection systems. To create the dataset, we have used Cooja emulator to establish a WSN, then, we have run multiple simulations with different configurations to generate both benign and malicious traffic.

TABLE I. PERFORMANCE METRICS AFFECTED BY EACH ATTACK

	Delay	Control Packets	Lost Packets	Received Packets	Sent Packets	Power Consumption	Packet Delivery Ratio	Routing Table Length	Protocol Settings
Rank attack	✓		✓	✓	✓	✓	✓	✓	
Sinkhole attack	✓		✓	✓	✓	✓	✓	✓	
Flooding attack	✓	✓	✓	✓	✓	✓	✓		✓
Copycat attack	✓	✓				✓			✓
Sybil attack	✓		✓	✓		✓	✓		✓
Wormhole attack	✓		✓	✓		✓	✓		

TABLE II. SELECTED PERFORMANCE METRICS AND DERIVED FEATURES

Performance Metric	**Features derived and description**
Delay	Average end to end delay (E2E)
Control Packets	Number of control packets transmitted through the network (Overhead). Presented as number of (DAO),(DIO) and (DIS) packets in RPL.
Lost Packets	The number of lost packets defined as (total sent packets – total received packets)(Lost)
Received Packets	The total number of packets received by the sink (Received), maximum number of packets received by a node (Max_received)
Sent Packets	The total number of packets sent by all nodes (Sent), minimum number of packets sent by each node (Min_sent), maximum number of packets sent by each node (Max_sent), minimum number of packets forwarded by each node (Min_forwarded), maximum number of packets forwarded by each node (Max_forwarded), total forwarding operations within the network (Forwarded).
Power Consumption	Total remaining power (Rem_power), maximum power consumption (Max_power), total power consumed by all nodes of the network (Total_power), average voltage of all sensors (Voltage)
Packet Delivery Ratio	The percentage of total received packets by the sink to the total packets sent by all nodes (PDR)
Routing Table	The maximum length of routing tables in all node (Max_length)
Protocol Settings	We have considered the beacon interval (Beacon), which is a varied period of time defines the frequency by which control packets are sent continuously by nodes in WSN.
Delay	Average end to end delay (E2E)

While benign traffic is generated by simulating a network with the original protocol, malicious traffic is generated by implementing a number of specified cyber-attacks. Then, the malicious copy of routing protocol is implemented and used to launch these attacks. The resulted traffic is collected and analyzed. Finally, features presented in Table I are extracted to form the final dataset. It is worth mentioning that the variations some features are considered instead of the mere features shown in Table I only. For instance, variations of sent packets are considered, such as minimum sent packets, and maximum sent packets for sent feature. Maximum power consumption, total power consumption and remaining power are considered instead of taking only average power consumption. Moreover, types of control packets are considered separately instead of counting the total control packets. We have included DIS, DIO, DAO control packets which are the main control packets in RPL protocol [30]. We have also considered forwarded packets as a special case of sent packets, which indicates packets received from neighbor nodes and sent again toward the intended destination. Table II presents selected performance metrics and features derived (between brackets).

D. Simulation Environment

Through this section, we present simulation that we have conducted and its related environment and tools. We have selected Cooja simulator to create and configure WSNs in a smart city. There are many reasons behind selecting Cooja simulator. Cooja simulator is designed specifically for WSNs, it implements Contiki operating system. Strictly speaking, a simulated sensor in Cooja presents an actual compiled Contiki system [31]. Moreover, Contiki, which is the operating system of sensors, is also the best candidate for IoT devices in a smart city. That is due to Contiki's design which is developed specifically for memory constrained devices with the consideration of low-power IoT. Existing employment of Contiki involves street lighting systems, radiation monitoring systems, sound monitoring systems and alarm systems [31]. As a result, Cooja simulator with Contiki operating system is the best choice to simulate the heterogeneous WSNs in a smart city system. According to the aforementioned points, resulted traffic will be similar to a great extent with a real traffic generated from real WSN rather than being just a traffic generated from a simulation.

Table III shows simulation environment and configuration parameters selected to create the dataset. Each record of the generated dataset represents 110 seconds traffic of 50 nodes deployed within an area of 350 X 350 m^2. At each record, we have diversified network configuration, such as the distribution of nodes, nodes to sink allocation and network topology. Part of the generated dataset instances represents benign network behavior, while the remaining dataset instances represent a traffic with the discussed attacks launched. After the simulation has been conducted for the 180 instances, performance has been measured in terms of the aforementioned performance metrics which are considered as dataset features to be inputs to the classification algorithm. Table IV presents a part of the generated dataset.

TABLE III. SIMULATION ENVIRONMENT AND PARAMETERS

Simulation Parameter	**Value**
Network Size (Number of Sensor Nodes)	50
Routing Protocol	RPL
Transport Layer Protocol	UDP
MAC Protocol	CSMA
Sensor Type	Sky Mote
Terrain Area	350 X 350 m^2
Number of Attacking Nodes	2
Simulation Duration	110 Seconds

TABLE IV. GENERATED DATASET

Overhead	**E2E**	**Sent**	**Received**	**PDR**	**Lost**	**Table**	**Max forwarded**	**Min forwarded**	**Max received**	**Min Sent**	**Max sent**	**Forwarded**	**Max power**	**Total power**	**Beacon**	**Rem power**	**Voltage**	**Class**
268	492.65	20	15	0.75	5	6	1	0	5	1	1	22	64360	964700	121.1	5040	252	normal
355	677.882	19	14	0.73	5	12	1	0	11	0	1	35	67447	799530	115.4	4284	252	normal
402	840.211	20	15	0.75	5	11	1	0	11	1	1	49	64080	930343	113.6	4788	252	normal
395	881.947	19	15	0.78	4	9	1	0	9	0	1	44	67897	949670	113.6	4788	252	normal
346	696	20	17	0.85	3	14	1	0	14	1	1	40	60265	837921	123.66	4536	252	attack
365	931.556	19	15	0.78	4	10	1	1	10	0	1	46	64914	895441	116.33	4536	252	attack
395	881.947	19	15	0.78	4	9	1	0	9	0	1	44	67897	949670	113.63	4788	252	attack
360	741.944	19	14	0.73	5	12	1	0	11	0	1	35	67441	874014	112.66	4536	252	normal
462	933.417	20	17	0.85	3	19	1	0	19	1	1	55	65172	637302	115.91	3024	252	normal
508	1199.23	20	16	0.8	4	18	1	0	17	1	1	75	69111	684722	103.0	3276	252	attack
426	926.133	20	15	0.75	5	15	1	0	14	1	1	50	66941	832173	92	3780	252	attack
307	540.8	21	15	0.714	6	7	1	1	7	1	2	31	66780	1059204	117.8	4723	248.75	normal

E. Machine Learning

We have utilized ML as a tool to set up inference rules. We have used Weka 3.8.4 environment to apply ML on the dataset discussed in Section A [32]. Decision tree (J48) classifier has been used as it is easy to extract the model in the form of explicit rules by this classifier. Decision tree classifier is trained and tested based on 10-fold cross validation technique.

V. Results and Discussions

By applying decision tree classifier on the discussed dataset, we have obtained 95% accuracy to classify a network traffic to either attack or normal traffic. We also have an accuracy of 81% to specify the name of the attack. The resulted rules then transferred to the ontology to carry out reasoning. Inference rules are shown in the table of Appendix B. Each rule in the table is represented in one row where the first column can be understood as a logical if statement and the second column represents the part of then statement.

A. Safety Factor

According to the previous accuracy values obtained, 95% for binary classification and 80% for attack type classification, 5% and 20% of cases are wrongly classified in both binary classification and attack type classification, respectively. However, these values cannot be considered as indicators for how much the system is secure. For instance, if the detection method wrongly classifies normal traffic as malicious, this can only add an additional cost, which is presented in taking a countermeasure reaction, but the system is still safe.

On the other hand, when the detection method classifies an attack traffic as normal, we can say that the system is not safe, because this means that we allow malicious behavior to proceed without being detected.

Fig. 7 presents confusion matrices for both binary and attack classifications. In binary classification, Fig. 7 shows that six cases out of 180 are malicious traffic that classified as normal, so, 174 cases are either correctly classified cases or they are normal traffic classified as attacks, which do not degrade the safety of a system. This means that the safety of the system in binary classification is 97%. In attack type classification, only four instances of attacks are classified as normal, which means that the safety of attack classification is 97.7%.

B. Use Cases

In this section, we introduce a simple use case to illustrate the validity of CAO. The ontology is implemented and rules obtained from ML model are added. Thus, we are ready to detect the occurrences of cyber-attacks using the proposed ontology. Different cyber-attacks are launched, traffic is analyzed to extract the defined performance metrics, then, calculated values are added and reasoning is conducted. Fig. 8 shows the description of copycat attack which has been defined as a network traffic using rules obtained from applying machine learning and in terms of performance metrics specified. Fig. 9 shows the definition of copycat attack in description logic.

We have simulated copycat attack, collected the resulted network traffic, analyzed the traffic to find out the values of performance metrics. We then defined the instance “traffic_1” of class traffic which have been assigned the obtained performance metrics as its data properties, and fed to the implemented ontology CAO. After reasoning is applied based on these inputs, an inference is done yielding that there is a copycat attack. The degree of accuracy of that inference is 95% in that there is an attack and 80% is that it is a copycat attack. The safety of using the proposed ontology along with the ML obtained rules is 97.7%.

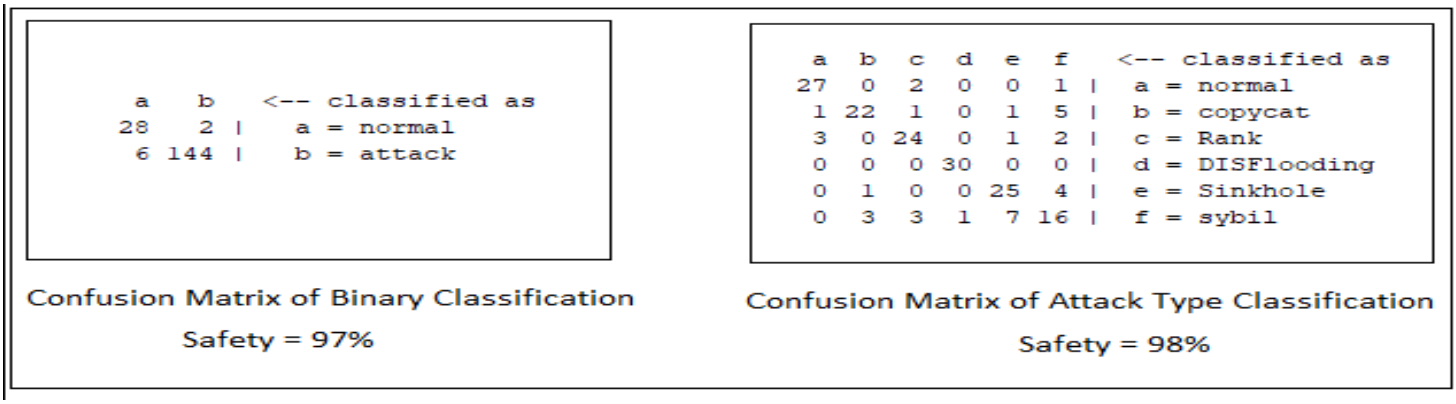


Fig. 7. Confusion Matrices of Both Binary and Attack Classification.

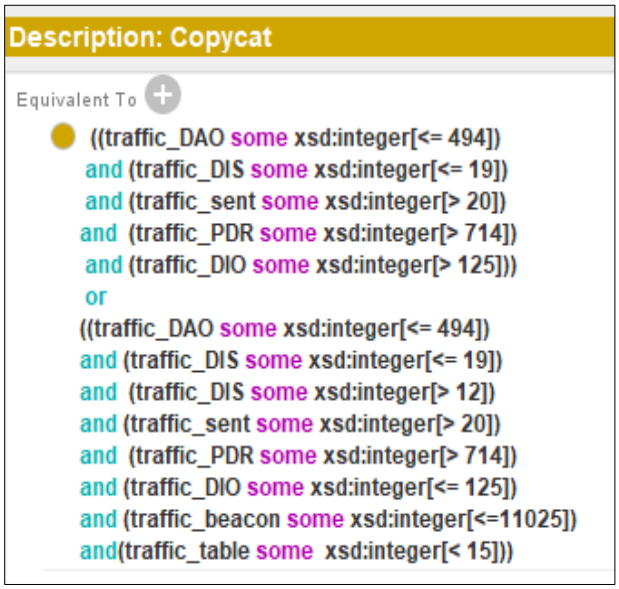


Fig. 8. Description of Copycat Attack in Terms of Traffic Specifications and Rule Extracted from Protégé Tool.

Copycat_Attack ⊑ Traffic ⊓ (<= 494 traffic_DAO) ⊓ (<= 19 traffic_DIS) ⊓ (>20 traffic_sent) ⊓ (>714 traffic_PDR) ⊓ (>125traffic_DIO) **OR** ((<= 494 traffic_DAO) ⊓ (<= 19 traffic_DIS) ⊓ (>12 traffic_DIS) ⊓ (>20 traffic_sent)) ⊓ (>714traffic_PDR) ⊓ (<=125traffic_DIO)) ⊓ (> 11025 traffic_beacon) ⊓ (>15 traffic_table))

Fig. 9. Definition of Copycat Attack in DL.

C. Comparison with Existing Works

In this section, we provide a comparison between CAO and existing research works investigating ontology and machine learning to secure smart city. Since there is no research work that have adopted the integration between ontology and ML to set inference rules of the ontology. We will compare our proposed approach in developing CAO with existing ontology-based approaches.

From the literature review in Section II, we have noticed that proposed ontologies in the domain of security are either environment-centric, assets centric, or threats-centric. However, the ontology developed in this work is cyberattack-centric ontology. Moreover, security needs in literature are considered from general view, which is not enough in IoT environment where security needs are considerably varied based on the application. In CAO, security needs are defined specifically based on the studied IoT system.

Finally, the main difference between CAO and previously proposed ontologies is the process of setting inference rules. Previous works have used the analysis of systems' vulnerabilities and threats, or they depend on user-defined rules. In CAO, we have benefit from ML to set inference rules.

VI. Conclusion and Future Work

In this article, we have proposed an ontology for cyber-attacks and we have shown how this ontology can be navigated to deduce the vulnerabilities, potential attacks, and the impact of the attack on the system in order to develop a model for an IDS for a smart city based IoT system. The proposed ontology is cyber-attack centric and includes other two main entities, adversary and system.

For future researches, CAO could be thought of as a base for more comprehensive ontologies, more concepts and relations could be added to develop cyber-attack oriented knowledge. Furthermore, CAO helps to develop a taxonomy for cyber-attacks. It also provides a formal and unified description and understanding for cyber-attacks and security needs for IoT environment.

We have investigated a case study for DoS attacks, six attacks have been considered and ML is used to set the rules for the proposed ontology. We have also proposed safety factor to evaluate the effectiveness of the IDS. The proposed ontology as well as the presented case study have shown that ontology can play a significant role to secure a smart city. The development of comprehensive ontology will establish a knowledge base for cyberattacks which creates the opportunity for robust protection for existing as well as coming security threats and attacks.

The proposed CAO needs to be supported by employing the proposed approach in a diversified set of computing environments, and investigating more types of cyber-attacks. Obtained accuracy values using CAO are less than detection accuracy in existing ML learning based security methods. The accuracy can be raised by studying several types of attacks or by the generating larger dataset which is the suggested future work.

REFERENCES

[1] Wang, D., Lee, S., Zhu, Y., & Li, Y. (2017, March). A zero human-intervention provisioning for industrial IoT devices. In 2017 IEEE International Conference on Industrial Technology (ICIT) (pp. 1171-1176). IEEE.

[2] Asassfeh, M., Obeid, N., Almobaideen, W. (2020, Dec), Anonymous Authentication Protocols for IoT based-Healthcare Systems: A survey, International Journal of Communication Networks and Information Security, 6(3), pp.302-315.

[3] Haque, A. B., Bhushan, B., & Dhiman, G. (2022). Conceptualizing smart city applications: Requirements, architecture, security issues, and emerging trends. Expert Systems, 39(5), e12753.

[4] Xu, G., Cao, Y., Ren, Y., Li, X., & Feng, Z. (2017). Network security situation awareness based on semantic ontology and user-defined rules for Internet of Things. IEEE Access, 5, 21046-21056.

[5] Almobaideen, W., Jarboua, H., Sabri, K.E. (2020), Searchable encryption architectures: survey of the literature and proposing a unified architecture, International Journal of Information Privacy, Security and Integrity, 4(4), pp. 237-260.

[6] Gopalan, S., Ali Raza, A., Almobaideen, W. (2021), IoT Security in Healthcare using AI: A Survey, 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), pp. 1-6, IEEE.

[7] Gheisari, M., Najafabadi, H. E., Alzubi, J. A., Gao, J., Wang, G., Abbasi, A. A., & Castiglione, A. (2021). OBPP: An ontology-based framework for privacy-preserving in IoT-based smart city. Future Generation Computer Systems, 113, 1-13.

[8] Gheisari, M., Pham, Q. V., Alazab, M., Zhang, X., Fernandez-Campusano, C., & Srivastava, G. (2019). ECA: an edge computing architecture for privacy-preserving in IoT-based smart city. IEEE Access, 7, 155779-155786.

[9] Alkhammash, E. (2020). Formal modelling of owl ontologies-based requirements for the development of safe and secure smart city systems. Soft Computing, 24(15), 11095-11108.

[10] Qamar, T., & Bawany, N. Z. (2020). A Cyber Security Ontology for Smart City. International Journal on Information Technologies & Security, 11(3), 63-74.

[11] Choi, Chang, and Junho Choi. "Ontology-Based Security Context Reasoning for Power IoT-Cloud Security Service." IEEE Access 7 (2019): 110510-110517.

[12] Mozzaquatro, Bruno Augusti, et al. "An ontology-based cybersecurity framework for the internet of things." Sensors 18.9 (2018): 1-20.

[13] Tao, M., Zuo, J., Liu, Z., Castiglione, A., & Palmieri, F. (2018). Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes. Future Generation Computer Systems, 78, 1040-1051.

[14] Oltramari, A., Cranor, L. F., Walls, R. J., & McDaniel, P. D. (2014, November). Building an Ontology of Cyber Security. In STIDS (pp. 54-61).

[15] Doynikova, E., Fedorchenko, A., & Kotenko, I. (2019, August). Ontology of metrics for cyber security assessment. In Proceedings of the 14th International Conference on Availability, Reliability and Security (pp. 1-8).

[16] Lannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R. & Goodall, J. (2015, April). Developing an ontology for cyber security knowledge graphs. In Proceedings of the 10th Annual Cyber and Information Security Research Conference (pp. 1-4).

[17] Salem, M. B., & Wacek, C. (2015). Enabling New Technologies for Cyber Security Defense with the ICAS Cyber Security Ontology. In STIDS (pp. 42-49).

[18] Obrst, L., Chase, P., & Markeloff, R. (2011, October). Developing an Ontology of the Cyber Security Domain. In STIDS (pp. 49-56).

[19] Syed, Z., Padia, A., Finin, T., Mathews, L., & Joshi, A. (2016, March). UCO: A unified cybersecurity ontology. In Workshops at the thirtieth AAAI conference on artificial intelligence.

[20] Bergner, S., & Lechner, U. (2017). Cybersecurity Ontology for Critical Infrastructures. In KEOD (pp. 80-85).

[21] Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on ML: How do IoT devices use AI to enhance security?. IEEE Signal Processing Magazine, 35(5), 41-49.

[22] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). ML in IoT security: current solutions and future challenges. IEEE Communications Surveys & Tutorials.

[23] Alghanam, O. A., Almobaideen, W., Saadeh, M., & Adwan, O. (2023). An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Systems with Applications, 213, 118745.

[24] Musen, M. A. (2015).The protégé project: a look back and a look forward. AI matters, 1(4), 4-11.

[25] Almobaideen, W., Allan, M., Saadeh, M. (2016), Smart archaeological tourism: Contention, convenience and accessibility in the context of cloud-centric IoT, Mediterranean Archaeology & Archaeometry, 16(1).

[26] Hashim Raza Bukhari, S., Siraj, S., & Husain Rehmani, M. (2018). Wireless sensor networks in smart cities: applications of channel bonding to meet data communication requirements. Transportation and Power Grid in Smart Cities: Communication Networks and Services, 247-268.

[27] Haq, N. F., Onik, A. R., Hridoy, M. A. K., Rafni, M., Shah, F. M., & Farid, D. M. (2015). Application of machine learning approaches in intrusion detection system: a survey. IJARAI-International Journal of Advanced Research in Artificial Intelligence, 4(3), 9-18.

[28] Almseidin, M., Alzubi, M., Kovacs, S., & Alkasassbeh, M. (2017, September). Evaluation of machine learning algorithms for intrusion detection system. In 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY) (pp. 000277-000282). IEEE.

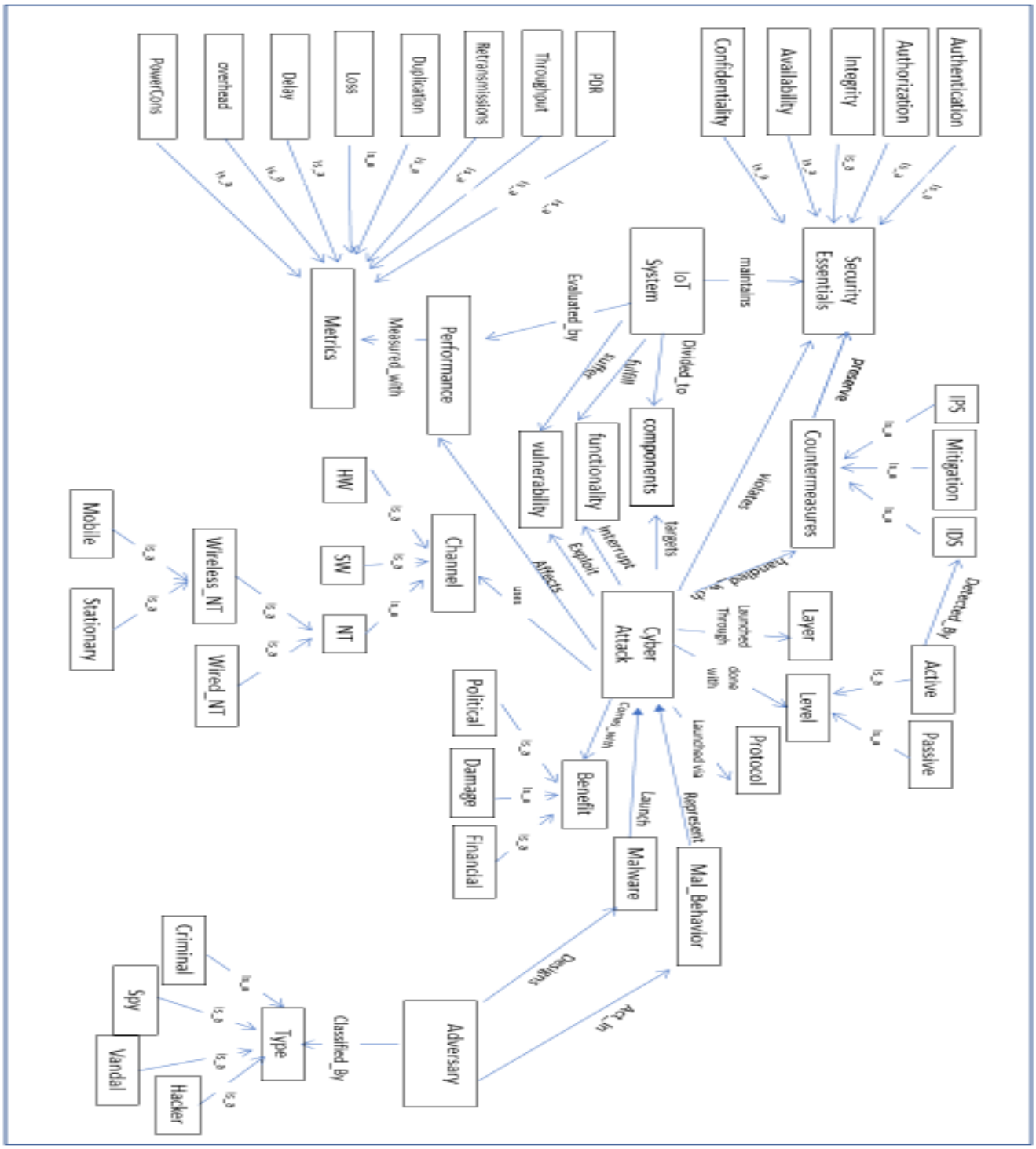
[29] Jain, A., & Jain, S. (2019). A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT. In Emerging Technologies in Data Mining and Information Security (pp. 611-620). Springer, Singapore.

[30] Zhang, T., & Li, X. (2014, August). Evaluating and Analyzing the Performance of RPL in Contiki. In Proceedings of the first international workshop on Mobile sensing, computing and communication (pp. 19-24).

[31] Padmaja, P. L., Ramanjaneyulu, T., Narayana, I. L., & Srikanth, K. (2017). Role of COOJA simulator in IoT. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 6(2), 139-143.

[32] Srivastava, S. (2014). Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining. International Journal of Computer Applications, 88(10).

APPENDIX A



Ontology Graph.

APPENDIX B

LOGICAL INFERENCE RULES

Rule	**Classification**
DIS <= 19 AND sent <= 20	Normal
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO > 115	Copycat Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon<=110.57 AND Table <=15 AND DIS > 11	Copycat Attack
DIS > 19	Flooding attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR <= 0.714 AND Rem_power> 3724 AND Min_forwarded =0	Rank Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR <= 0.714 AND Rem_power> 3724 AND Min_forwarded > 0 AND received<=11	Rank Attack
DIS <= 19 AND sent >20 AND DAO>494	Sinkhole attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR <= 0.714 AND Rem_power<=3724 AND Min_sent =0	Sinkhole attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon>110.57 AND Min_sent > 0 AND PDR <=0.77	Sinkhole attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon>110.57 AND Min_sent > 0 AND PDR >0.77 AND Max_power > 68598 AND DIO<=113	Sinkhole Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR <= 0.714 AND Rem_power<=3724 AND Min_sent > 0	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR <= 0.714 AND Rem_power> 3724 AND Min_forwarded > 0 AND received>11	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon<110.57 AND Table >15	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon<=110.57 AND Table <=15 AND DIS <=11	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon>110.57 AND Min_sent = 0	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon>110.57 AND Min_sent > 0 AND PDR >0.77 AND Max_power <=68598	Sybil Attack
DIS <= 19 AND sent >20 AND DAO<=494 AND PDR > 0.714 AND DIO <= 115 AND Beacon>110.57 AND Min_sent > 0 AND PDR >0.77 AND Max_power > 68598 AND DIO>113	Sybil Attack