Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Information Networks

An Ontology – Based Semantic Firewall Rules for Network Protection

A Thesis
Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology-Information Networks

By

Baraa Hashim kareem **Fejer**

Supervised By

Prof. Dr. Wesam Sameer AbdAli Bhaya

2021 A.D. **1443 A.H.**

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿ قَالُوا۟ سُبْحَٰنَكَ لَا عِلْمَ لَنَآ إِلَّا مَا عَلَّمْتَنَآ ۖ إِنَّكَ أَنتَ ٱلْعَلِيمُ ٱلْحَكِيمُ ﴾

صدق الله العظيم

سورة البقرة

آية (٣٢)

DEDICATION

I dedicate my work to...

(My father)

To the great man who offers his life for us to get to where we are now and I will not disappoint you, my beloved father.

(My Mother)

The guarding angel, the pure affection, one woman without her I would never be what I am, who provides me with love, strength, and courage, the person to whom I am still indebted, the dearest person.

(To all my sisters and brothers)

Who stand by me when things look very difficult.

(To My dear husband and my beautiful daughter)

my loved ones in the march of life from whom I gather my strength.

I would like to express my appreciation and gratitude to (Dr. Mohammed Hussein Naji).

And finally, my supervisor Prof. Dr. Wesam S. Bhaya, lead me all the time towards lights of hope.

Acknowledgment

In the name of Allah, the Most Gracious, the Most Merciful

At first, the greatest praise is to Allah for assisting me to face the difficulty that I met throughout my study, and for always helping me to achieve my aims.

I owe a deep debt of gratitude to the College of Information Technology and the University of Babylon for offering me the opportunity to complete this work.

I would like to express my sincere thanks and appreciation to my supervisor (Prof. Dr. Wesam S. Bhaya) who supported me from the beginning until the completion of the current thesis. I greatly appreciate his optimistic behavior, valuable advice, and trust in me that have always encouraged me to complete this work.

I also would like to express my wholehearted thanks to my family for their unlimited support, encouragement, love and great sacrifice they provided me with throughout my life.

Declaration

hereby declare that this dissertation entitled " **An Ontology – Based Semantic Firewall Rules for Network Protection** ", submitted to University of Babylon in partial fulfillment of requirements for the degree of Master in Information Technology \ Information Networks, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source are appropriately cited in the references.

Signature:

Name: Baraa Hashim Kareem

Date: /   / 2022

Abstract

The amount and variety of network attacks keep growing while the same basic defense attack techniques are being used. A firewall is a network security key component that filters inbound and outbound network packets as per predefined security rules.

Even though the firewalls are an effective defense against some attacks, they have certain security flaws that can be leveraged in other circumstances. The old firewall can bypass a network intrusion anomaly. For anomaly network intrusion in the new firewall, it is claimed that an ontology-based semantic firewall and machine learning algorithms can effectively enhance the firewall and protect the network. This thesis proposed an ontology-based model as a semantic firewall as an effort to explore its effectiveness.

The method proposed in this thesis is based on Description Logic Reasoners (DLR) and Ontology APIs with Semantic Web Languages and Semantic Web Rule Language (OWL and SWRL) supported with machine learning algorithm. The proposed semantic firewall takes its decisions of anomalies detection based on a set of protection rules of the ontology-based model. To achieve this, network intrusion detection dataset is fed to a set of machine learning classification algorithms including Naive Bayes, k-nearest neighbors, Decision Tree, Stochastic gradient descent, and logistic regression. Accuracy is used for performance evaluation. In the experiments, Decision Tree has shown the best performance.

The decision tree algorithm is used for calculating the threshold that is the backbone of rules-based reasoners of SWRL that are used in the detection of anomalies. Then, a model based on ontology was built on 30 semantic firewall rules. Based on these rules, the reasoner can give the decision.

As a result, the proposed approach achieves a detection accuracy of 95%. Finally, the conclusion is drawn that the ontology classifier gives a comprehensive model of the semantic firewall which provides candid and human-interpretable decision rules, against the alternative machine learning models.

Table of Contents

List of Figures

List of Tables

Table of Abbreviations

Abbreviations	The Description
API	Application Programming Interface
OWL	The Web Ontology Language
SWRL	The Semantic Web Rule Language
SFW	Semantic Firewall
ML	Machine Learning
RDF	Resource Description Framework
RDF-S	RDF Schema
SPARQL	SPARQL Protocol and RDF Query Language
HTML	HyperText Markup Language
XML	Extensible Markup Language
UNICODE	Universal trunk - out-of-service CODE
URIs	Uniform Resource Identifiers
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
SW	Semantic Web
DL	Description Logic
DT	Decision tree
DLR	Description Logic Reasoners
NB	Naive Bayes
KNN	k-nearest neighbors
DT	Decision Tree
SGD	Stochastic gradient descent
LR	Logistic regression

Chapter One

General Introduction

1.1 Introduction

Firewalls are considered to be a highly robust security guard which has become an integral element of almost every network. It was originally thought to be found between two networks. Yet, as the usage of smaller networks and internet expanded, it became one of the most important components of any gateway to prevent external intruders from entering other private networks and LANs. In nearly every business, firewalls are deployed as the first line of protection for network security. Firewalls are one of the most effective methods for ensuring the highest security level in computer networks [1]. The IT security community has not incorporated data mining techniques. This lack of implementation can be attributed to a variety of factors, such as a lack of comprehension of the approaches and the difficulty in acquiring the essential data. The obstacles to obtain data have been steadily reduced over time as network security professionals deal with network traffic [2]. The use of decision trees, a data mining tool, can help with this issue. Decision trees can facilitate technology-specific strategies to combat against assaults by providing valuable insights into the challenge of recognizing malicious activity [3]. The Semantic Firewall is pictured as a service that works in tandem with a traditional firewall to determine if incoming and outgoing messages are permissible. This is done in accordance with the security regulations placed inside the protected domain where the messages are transmitted or received. The system administrator is in charge of this device, and it is also considered to be accountable in setting the proper policies [4].

Ontology-based knowledge modelling is a viable option since it allows for the separation of meaning from processing, thereby enabling the creation of huge collaborative knowledge bases. Many disciplines use ontologies as a modelling tool because integrating and uniformly representing information received from diverse sources can be difficult. The Web Ontology Language (OWL) is recommended by the W3C for ontologies used to specify the web, but it is also widely used in other application areas [5]. Even though there is a widespread use of ontologies in several fields, ontology-based methods for semantic firewall and network administration remain uncommon.

In traditional firewalls, the rule policy is organized in a 'listed rule' list of rule lines. The listed rule can potentially lead to three major issues: speed, security, and user-friendliness. Many packets will be matched with the rule at the bottom places, which might cause speed issues. With numerous rules situated above the matched rule, the firewall may consume unnecessary time in validating packets. Furthermore, traditional firewalls encounter rule conflicts, such as shadowed rules. Many rules designed to prohibit attacking packets may be obscured by rules above them and therefore are unable to block any packets. This allows hazardous packets from the outside to penetrate within networks. Moreover, traditional firewalls lack user-friendly functionality since administrators must have adequate experience to set sufficient efficiency rules. When deciding on a packet, seeking for data in the tree rule is faster than searching for data in the classical firewall's listed rule. This is due to the fact that searching a Tree is quicker than sequentially gathering data in Arrays. In addition to that, rule conflicts can be eliminated since each packet is evaluated against the associated 'rule path' in the tree rule [6]. This work presents a semantic firewall by using a machine-learning and ontology-based model to solve the aforementioned problems.

The firewall rules are modeled with ontologies by deploying OWL to represent knowledge. The formal representation will enable the systematic disambiguation and structuring of the described knowledge. The Semantic Web Rule Language (SWRL) is used to validate policies, such as identifying firewall problems.

1.2 Related Work

The following sections present the most prominent two approaches: a machine learning techniques and semantic modelling techniques. Researchers have analyzed firewall attacks detection based on a semantic approach and ontology engineering.

M. H. Jabardi et al, 2021 presented for integrating the two paradigms The Semantic Web and Machine Learning and used data from Twitter regarding legitimate and fake accounts. SWRL rules-based reasoner is utilized under predefined rules to infer whether the profile is trust or fake. It was achieved a high detection accuracy of 98%. Used decision tree algorithm for extract the rules. Furthermore, ontology classifier is an interpretable model that offers straightforward and human-interpretable decision rules [7].

Q. Ismail et al, 2020, are utilized a dynamic ontology of different firewall rules managed by SPARQL queries, so that the rules are applied faster, and thus, increasing the firewall performance. Experimental results show that proposed methodology totally eliminates the anomalies in the firewall rules as a result of conducting longest matching with proper rules from the dynamically constructed ontology. used 3000 packets as sample to check the accuracy of the ontology-based firewall, was 100% [9].

Hajar, H. et al, 2020, The proposed methods have shown a more superior and precise performance in terms of anomaly detection accuracy and as a result, enabling network administrators to update and optimize Firewall policy rules. The training data have been divided to 6 pieces 20-50-100-200-300-500 thousand records. We have compared the performance of the model based on four classification algorithms, NaiveBayes, KNN, One'R, and J48. It was noticed that, KNN algorithm has showed better successful performance results than the other 4 classification algorithms. the accuracy of the proposed model is around 100% [10].

R. F. Cordova et al, 2018, presented an efficient ontology design for detecting misconfigurations on firewall rules, that attempts to reduce the computing resources needed to validate the firewall rules of the company’s policies. The design was tested on a real-world scenario of an enterprise with equipment from 3 different vendors: Fortinet, Cisco ASA, and Checkpoint. Our solution was able to detect over a hundred misconfigured rules. Finally, an evaluation of the impact of the chosen combination of ontology, query language, and reasoner on the computational cost is also presented [11].

J. Vincent et al, 2011, describes a proposed approach that involves semantic web languages (OWL and SWRL) and tools (DL reasoners and ontology APIs). The authorization/forbidding decisions are made based on certain privacy-related rules for two ontologies which respectively model the identification of mobile phone users and privacy policy. The aforementioned ontology-based approach is validated by means of a proof of concept which involves an accurate privacy-threatening scenario. This case has its implementation in Java, outlining the semantic firewall ported to the Android platform [12].

1.3 Problem Statement

The firewall can be targeted by a network intrusion detection anomaly. Despite the uses of firewalls in defending particular attacks, there are a number of security gaps that could be crossed. Security problem, caused by potential shadowed rules and the change of meaning of the rule policy due to rule repositioning. Difficulty in rule design, in which one needs to carefully choose the proper positions for the firewall rules in order to avoid listing rules.

1.4 Aim of thesis

1. To address the shortcomings of traditional firewall, ontology-based semantic firewall with machine learning algorithms can effectively enhance the firewall and protect the LAN.
2. The proposed model's scenario is the use of an ontology, along with semantic rules and reasoners for learning the rules which classify anomalies. In addition, an ontology classifier is used as a decipherable model with simple and human-interpretable decision rules.
3. Building a model that detects firewall attacks using machine learning to extract the threshold (the cutoff point).

1.5 Thesis Objectives

This thesis aims to present a new approach for detecting and classifying anomalies according to machine learning with ontological engineering and SWRL rules. The objectives of this work can be defined as follows:

1- Building a model that detects network intrusion attacks using machine learning to extract the threshold (the cutoff point) for each feature; these thresholds are the backbone of building semantic rules.

2- Building an ontology classifier for firewall attacks classification. Furthermore, an ontology is constructed for the representation of information, and an understandable model is designed to offer direct and human-interpretable decision rules.

3- Creating semantic rules depending on the Network intrusion relationships that is used to infer anomalies.

1.6 Thesis Organization

The thesis consists of five chapters, each of which begins with a short overview that offers a general impression of the chapter. The key contents of other chapters are stated below:

Chapter 2: The theoretical background introduces the central concept of semantic firewall ontology and the role it plays within the context of the Semantic Web technologies (RDF/S, OWL, SWRL, and SPARQL). The semantic web layered architecture is explained briefly. A detailed overview of the Web Ontology Language (OWL) is provided along with its various dialects. It provides necessary information about the reasoning engines. The uses of the Semantic Web Ontology Language (SWRL) rules ae described for inferring and distinguishing individuals of classes.

Chapter 3: This chapter presents the architecture of the proposed model for the Semantic Firewall and explains how the Semantic Web Ontology Language (SWRL) rules are used to infer and distinguish anomalies. The ontology is developed using Protégé and OWL. Other aspects discussed are the data pre-processing, and features selection.

Chapter 4: Implementation and results: This chapter demonstrates the results of the proposed model and the research experiments. It also discusses the evaluation of the system's performance.

Chapter 5: The last chapter states the research conclusions and suggests a number of possible future works.

Chapter Two

Theoretical Background

2.1 Introduction

The important and essential process in the Internet is how to transfer the data packets with higher security. Some security policies are applied by firewalls to satisfy the safety access requirements of network packets in blocked mode [18]. A firewall is a set of components that aims to protect the inner network against the unauthorized access attempts from outer network. The filtering manner depends on a set of rules called "security policies" that control the firewall management [19]. The firewall applies its rules on several types of attacks, like Denial-of-service, Eavesdropping, Host Attacks, Password Guessing, Protocol-based attacks, Social Engineering and War Dialing. Firewalls have important policies that manage firewall rules such as: Firewall policy editor, automated correction of policy fault, Fault localization, Anomaly detection and Policy modeling [20].

Firewalls can be represented as a barrier that separates Local Area Network (LAN) and the Internet. It keeps private resources confidential and keeps the security risks to a minimum. Figure (2.1) shows an example of a firewall between LAN and the internet. Both hardware and software could be applied in filtering the traffic of networks [21].

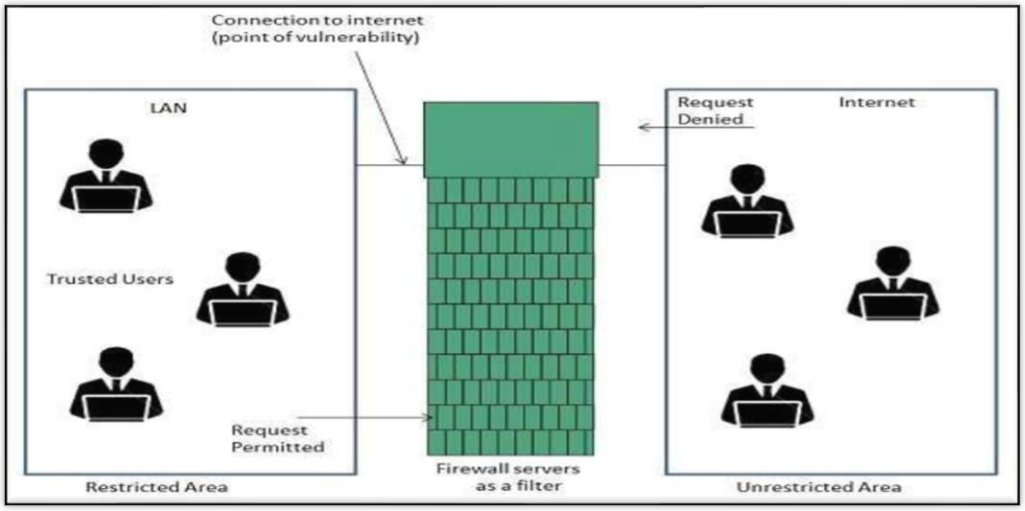


Figure (2.1) Firewall protections [21].

2.2 Firewall protections

Firewalls are an important part of the security of any network. Firewalls can be of two types namely, network firewalls or host-based firewalls. Network firewalls are those that help to provide security between networks and this run-on network hardware. Host-based firewalls are those that help to filter traffic to the end devices or the hosts.

Firewalls traditionally worked on rules that were configured statically like access policy lists. Over the years, with increasing threats they have evolved to dynamically detect and react to threats to the network. The different types of Firework technologies work at different layers of the TCP/IP model. The different types of firewall technologies [12]:

- Packet-filtering firewalls.
- Circuit-level gateways.
- Stateful inspection firewalls.
- Application-level gateways (proxy firewalls).
- Next-generation firewalls.
- Cloud firewalls.

2.2.1 Network Security with firewall

The increase in Internet use is accompanied by the increasing demand for Network Security. Network Security aims to prevent illegitimate admittance, hacking and authentic data transportation [13]. It involves a number of provisions and policies used by network administrators for precluding and monitoring the cases of unauthorized access, alteration, perverting, as well as the decline of computer networks and network-accessible resources. This is obtained through Firewalls.

Firewalls are hardware or software devices that are designed for permitting or refusing network transmissions according to particular protocols [14].

In principle, firewalls tend to inspect all traffic that occurs between two networks, making sure that all prototypes are met. The firewalls are routed between those networks whenever they match the prettified prototypes, otherwise it will be stopped. Firewalls do not only help to limit the access of undesired or malicious hosts or users, but it also dismays any upcoming threats. By refining both incoming and outgoing transit, it manages public access to private networked resources like host applications. A firewall has the ability to filter packets according to the addresses of the source and destination as well as the port numbers. This process is also known as address filtering. Another form of filtering is protocol filtering, whereby the firewall filters particular kinds of networks. This traffic can also be filtered based on the packet attributes or states. [15].

2.2.2 Types of Network Attacks

As for the different classes of attacks that the network might be subjected to, these involve the passive monitoring of communications, active network attacks, close-in attacks, exploitation by insiders, and attacks via the service providers. Information systems and networks represent a much-aimed target, and therefore need to be able to resist such forms of attacks that might occur from threat agents like hackers and nation states. Systems also need to restrict any damage, recovering it as rapid as possible in cases of attack. Some of the most commonly occurring attacks include [16]:

1. Passive Attacks
2. Active Attacks
3. Distributed Attacks
4. Insider Attacks
5. Close-in Attacks

6. Phishing Attacks
7. Hijacking attacks
8. Spoofing attacks
9. Buffering overflow
10. Exploiting attacks
11. Password attacks

2.3 Machine Learning

Machine learning (ML) refers to the concept that involves learning from past experience (previous data) for improving future performances. The main focus of this field is automatization of learning methods. Learning here means modifying or improving algorithms according to past “experiences” in an automatic way with no external human assistance. The design of a machine (software system) requires the programmer to define a particular purpose. In this case, ML is used to let the machine itself find a solution rather than having the researcher design an algorithm that addresses the issue in a direct way. The algorithm will reach its own solution through either an example or a training dataset that is provided in advance [17].

2.3.1 Types of Learning

There are three major kinds [18]:

1. Supervised Learning.

2. Unsupervised Learning.

3. Semi -supervised Learning.

Figure (2.2) shows the major machine learning kinds.

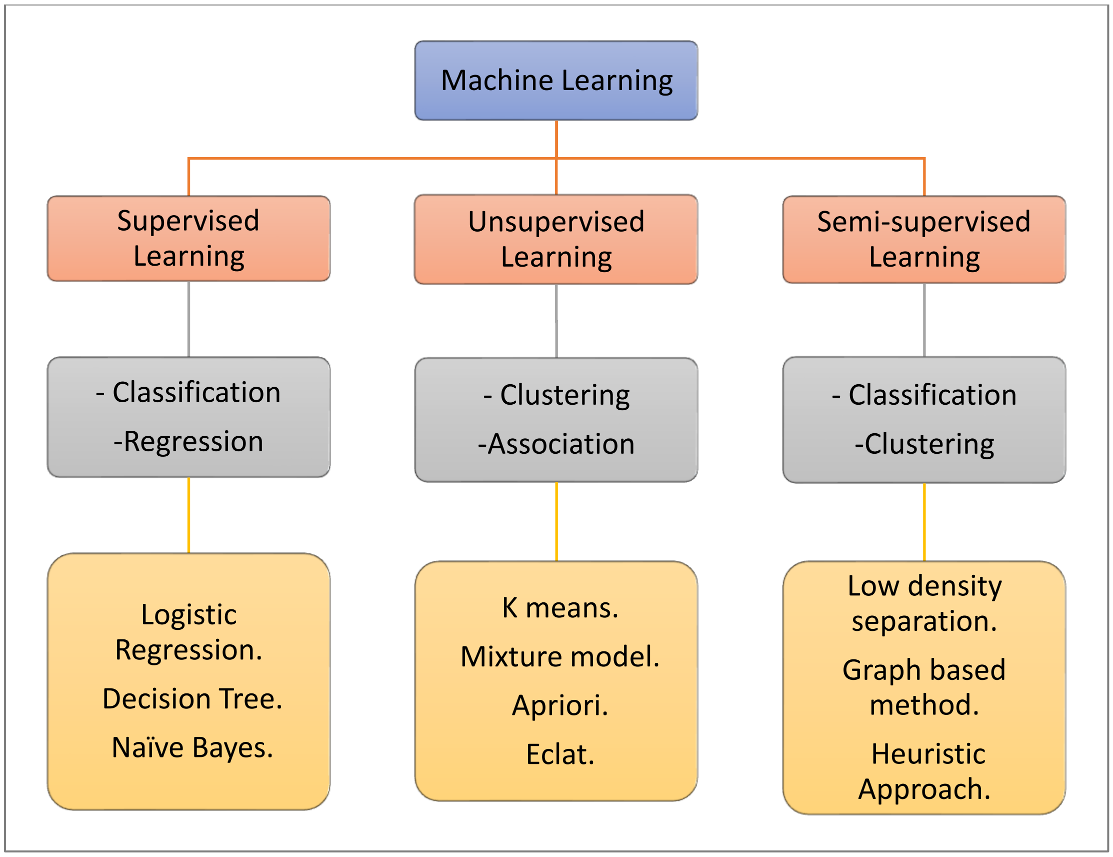


Figure (2.2) Major Machine Learning kinds [18].

1- Supervised Learning

This algorithm refers to the kind that needs extrinsic help, whereby the input dataset consists of two sets: training and testing. The first database has outputs that are changeable which necessitate to be expected or categorized. Whole models take learning of several types of algorithms from the training data base and implement them to the testing one for expectation or categorization [19]. The workflow of supervised machine learning algorithms is explained in Figure (2.3).

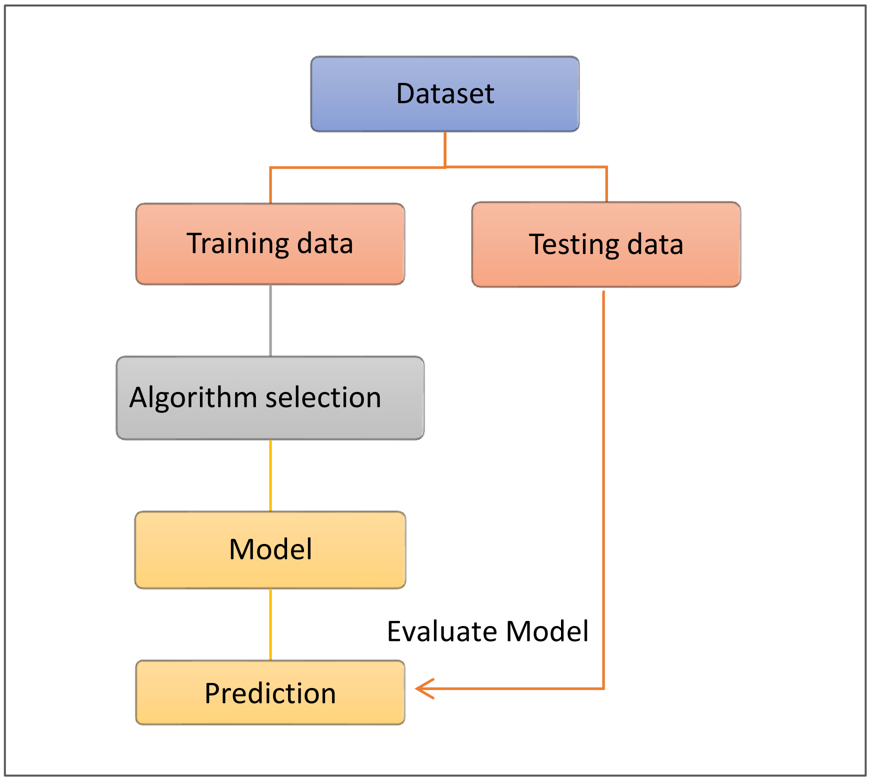


Figure (2.3) The Processes of Supervised Machine Learning [19].

2- Unsupervised Learning

This algorithm model learns a number of characteristics from the data. When new data is put in, it utilizes the prior learned characteristics to identify the class of the data. Mainly, it has a function of clustering and feature reduction [20].

3- Semi-supervised Learning

This model is a way that joins the power of supervised and unsupervised learning. That combines a small quantity of labeled data with a big quantity of unlabeled data during training. whereby the unrecognized data is already introduced and obtaining labeled data is a boring process [21].

2.4 Decision Tree Algorithm

Decision trees are one of the most commonly used ML algorithms applied to solve in a variety of classification problems. Besides classification, it has a number of other applications such as regression. Decision Tree is a tree-shaped diagram in the form of a ranching tree used to determine a type of action. Decision trees provides outputs that can be easily comprehended by people. It presents a tree-like diagram whereby the nodes, branches, and leaves represent attributes, rules, and target classes respectively. Target classes can be either separate or have no interruption. Decision trees may take the form of (IF-then-Else) rule. Bigger decision trees tend to require more complicated rules [22].

As for the structure of the decision tree, the inner nodes are tested on attributes. The branches are the outcomes of these tests, and the leaves or terminal nodes are the class labels. Provided that a tuple is X, its attribute values are tests in light of the decision tree. This is followed by tracing a path from the root to the leaves that hold the class predictions of the tuples. Decision trees can be easily converted into classifying rules. In decision tree learning, decision trees are used as prediction models for mapping observations of a particular item so as to conclude upon that item's target value. This is considered to be a commonly used modern predicting approach found in statistics, ML, and data mining [23].

Functioning on the basis of Hunt's algorithm, both categorical and continuous attributes are treated by J48 when building decision trees. For the first type of attributes, their values are divided into two parts according to a determined threshold, so that all values over the threshold are one child, and the rest represent another. Missing attribute values are also treated. The Gain rate is used as a measure to select attributes for building decision trees, as it removes the biasness of information gain in case of multiple outcome values. After calculating the gain ratio for every attribute, the root node will represent the attribute that has the highest gain ratio. The information gain of an attribute A is calculated as follows in Eq ($\mathbf{2.1}$):

$$\boldsymbol{gain} = \boldsymbol{info}(\boldsymbol{T}) - \sum_{i=1}^{s} \frac{|Ti|}{|T|} \times info(Ti) \quad \ldots\ldots\ldots\ldots\ldots\ldots (\mathbf{2.1})\ [24].$$

where, T is set of cases and Ti (i=1 to s) are the subsets of T consisting distinct value for attribute A. info (T) is known as entropy function described as follow in Eq ($\mathbf{2.2}$) :

$$\boldsymbol{info}(\boldsymbol{T}) = -\sum_{\boldsymbol{j=1}}^{\boldsymbol{N\ class}} \frac{\boldsymbol{freq}(\boldsymbol{Cj}, \boldsymbol{T})}{|\boldsymbol{T}|} \times \boldsymbol{log2}\left(\frac{\boldsymbol{freq}(\boldsymbol{Cj}, \boldsymbol{T})}{|\boldsymbol{T}|}\right) \ldots\ldots\ldots..(\mathbf{2}.2)\ \mathbf{[25].}$$

J48 makes use of pessimistic pruning when unnecessary branches within the decision tree are removed for improving how accurate the classifying process. In decision trees, all leaves have certain class labels assigned to them. As for the non-terminal nodes (roots and internal ones), they have particular attribute test conditions for separating records of differing features [26].

2.5 Data Preprocessing

Data pre-processing includes a number of techniques applied before executing machine learning methods. The elimination of noise instances is one of the most difficult problems in inductive ML. usually, the removed instances have excessively deviated instances that have too many null feature values. These excessively deviating features are also referred to as outliers. In addition, a common approach to cope with the infeasibility of learning from very large data sets is to select a single sample from the large data set. Missing data handling is another issue often dealt with in the data preparation steps [27]. It is a known problem that features with too many values are overestimated in the process of selecting the most informative features, both for inducing decision trees and for deriving decision rules [28].

2.6 Dataset Description

This section presents some background information about the Network Intrusion Data set that is used throughout the present work. The data set provides a varying range of intrusions that are mimicked within military network environments, given for auditing. By replicating a typical US Air Force LAN, it provided a context in which raw TCP/IP dump data of networks could be obtained. The LAN was targeted as if it were a real setting, and several attacks were launched. Connections are a series of TCP packets which begin and terminate at a particular time interval and allow data to flow from the source IP address towards the target IP address using a precise protocol. In addition, these connections are classified into being either normal or attacks, having only a single form of attack. All connection records are around 100 bytes long. Out of the normal and attack data, 41 quantitative and qualitative characteristics are extracted for every TCP/IP connection (38 quantitative features and 3 qualitative). The dataset consisted of (25192) rows and (42) columns The class variable has two categories [29]:

- Normal
- Anomalous

2.7 Semantic Web Technologies

Tim Berners-Lee was the first to mention the concept of Semantic Web, introducing it at his key-note speech during the first World Wide Web conference in 1994 [30]. Iteratively developing ontologies in the stream of environment data indicates that the SW technology is important when addressing two major issues:

- The interdependency between disparate data sets. This could be solved through the semantic enrichment of low-level sensor measures of ontologies and reasoning over the resultant enriched data sets to derive new knowledge.

- The geospatial data integrating and reasoning problem. This is solved through the semantic enrichment of all sensor measures by means of the utilization of ontologies and interoperable metric unit's conversion. All sensor measures assigned the associated metric units in a semantic manner through the ontology to translate it via the inference rules. The total ontology also contributes to this aspect as it provides a proof of concept of the manner through which ontologies could address the needs of certain given environment projects, whenever the stream of data for environmental IoT [31] deployed in North Wales.

In SW, there are a number of technologies, tools, and standards identified that create the fundamental building elements of its infrastructure, for supporting the associated Web with meanings. The SW architecture consists of a set of standards ordered in form of particular structures to express the existing inter-relationships. The representation of the architecture often involves the use of the diagrams initially introduced by Tim Berners-Lee in 2001 [32].

2.7.1 The Semantic Web Layer Architecture

The semantic network layer stack is an example of the information hierarchy whereby each layer utilizes the bottom layers' tools to perform its job. It shows how technology, principles, and resources are designed to make the semantic Web possible. Tim Berners-Lee, the creator of the World Wide Web as well as SW, introduced four versions of the SW layers stack during the period from 2000 to 2006 (V1 to V4), as shown in Figure (2.4). The stack layer updating is still progress with the layers' concretization [33] [34].

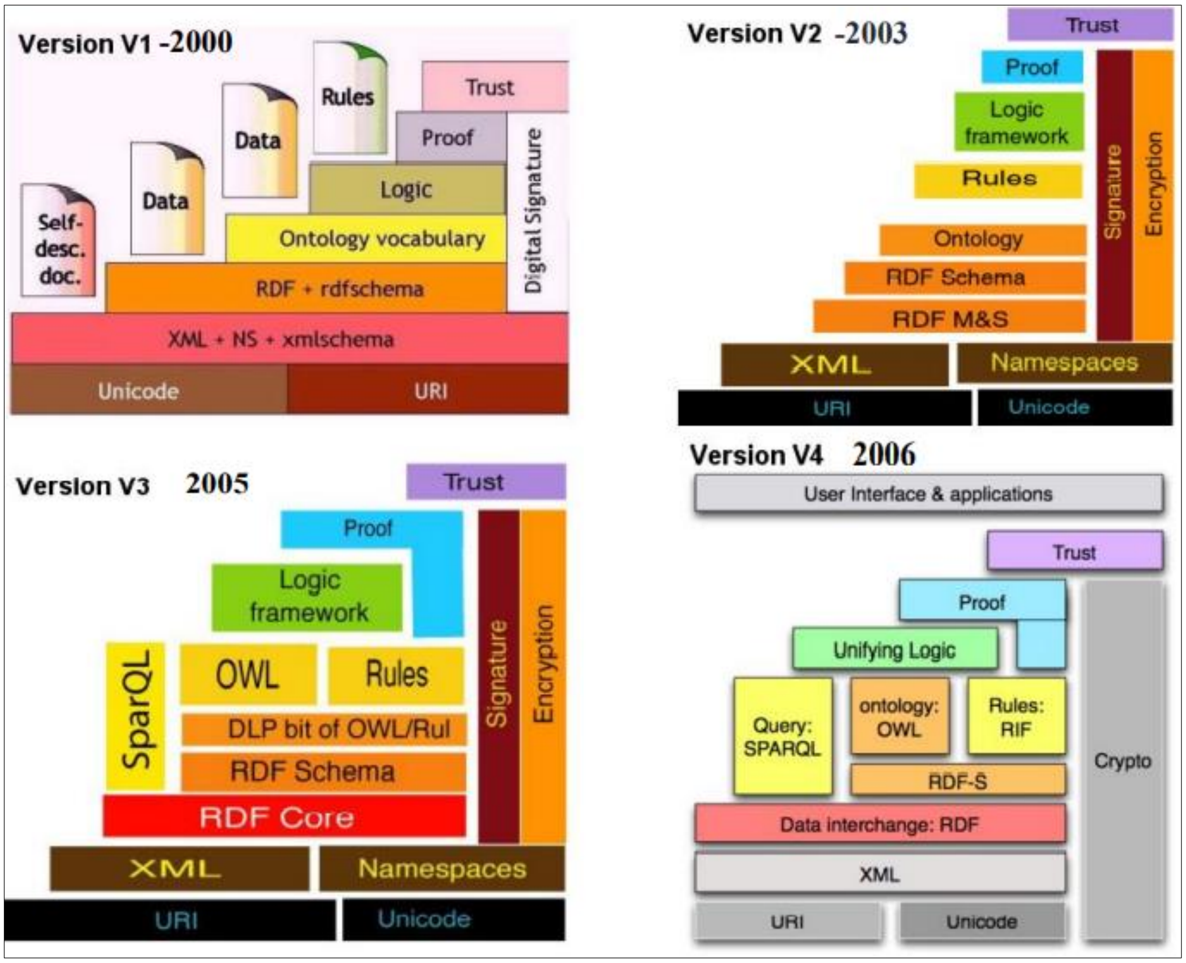


Figure (2.4) The Different Semantic Web Architectures (V1-V4) proposed by Tim Berners-Lee [34].

The main components are organized in a layered architecture into a stack whereby every layer exists above a lower layer. In general, a layer describes a set of components which provide associated tasks. A top layer can either use different services specified by the immediate bottom layer or use all bottom layers' services. However, the bottom layers are ignorant of the upper layer and do not have access to features from the higher layers [35] [36]. In a unidirectional layered architecture, access to the components' services is strict of order.

In the **first layer (Identity layer)**, the URI uniquely identifies semantic web resources while the Universal trunk - out-of-service CODE (UNI CODE) is used for representing text in a different language [37].

In the **second layer (Syntax & Structure layer)**, Extensible Markup Language (XML) used to interchange structured data over the Web while various mark-up vocabularies are specified within a single XML document. This is done with the help of XML name space [38].

In the **third layer (Semantic layer)**, the Resource Description Framework (RDF) represents knowledge of resources for semantic Web graphically. The RDF Schema could be applied in describing the taxonomy of classes and characteristics so as to be used in creating light-weight ontologies [39].

As for the **fourth layer (Ontology layer)**, Web Ontology Language (OWL) is utilized to create ontologies. Ontologies are the essential elements of the semantic Web. These are documents that describe the terms' relationships. Ontologies aid with classifying data and information as groups or taxonomies. A Simple Protocol and RDF Query Language (SPARQL) can be applied for the direct query of an ontology or knowledge base. SWRL rules and reasoners can be adopted for inferring novel knowledge from what already exists [40].

The **fifth layer (Logic layer)** supplies a vocabulary to explain the validity or falsity of claims that are made. In this layer, the FOL uses enhanced RDF statements with richer syntax and more semantic power. The logic layer is used for reasoning-based interaction about data/information merging [41].

In the **Sixth layer (proof layer)**, the Proof layer exposes the reasoning steps that led the Logic layer to make the inference that are made. The Proof layer's closest implementation experiences seem to fall into formal methods for proving programs correct and automated theorem proving, confirming the connection between queries and content, using inference, and providing a structure of explanations [42].

The **last layer (Authentication layer)**, is responsible for creating a level of trust between the user and the lower layer's information quality. Trust is connected to verifiable statements concerning identity, the provenance of the information, and relevant issues. The basic definition of identity, preserving the mercurial condition of confidence, remains threatened [43]. Current trust tools such as certificates of authentication protect the exchange of information, and the algorithm combination of trust relationships reflect a wide range of trusted exchange of information [44].

2.7.2 Uniform Resource Identifier (URI) and UNICODE

URI and Unicode represent the first layer of the SW stack layers. URI is a single universal identification method used to assign unique names to resources-from abstract concepts and web content to real-world entities. A URI aims to describe resources (classes, properties, and entities) clearly and Regularly [45]. Uniform Resource Locator (URL) is a sub-set of URI which includes a data access technique and location. For example, the URL for the suggested ontology's home page is:

http://www.semanticweb.org/Baraa/ontologies/2021/5/SFW

Unicode is an international code set encoding standard that allows the use of a standardized pattern by all human languages on the Web. Regardless of device, platform, software, or language, the Unicode Standard offers a unique number. Thus, sending or receiving data without damage across various networks, machines, or applications. Unicode makes conversation between devices easily [46].

2.7.3 Resource Description Framework (RDF)

Hypertext Markup Language (HTML) is the default language to write web pages in such a way that it is understandable by a human. It allows users to publish a document and assures them that any web browser could render it. SW needs something more potent, and it needs a data model that can be used by different applications, not only to explain documents for people but also to understand machines and be independent of the domain. The Resource Description Framework (RDF) provides such a flexible domain. RDF is a leading data description format for the semantic Web. RDF was primarily designed to reflect metadata about the Site resources. RDF represents statements based on triples format (subject, predicate, value) and a related XML syntax. URIs must represent a subject and predicate, while an object can be a static data value or a URI [47].

2.8 Identity Ontologies

There are many contradictory definitions of ontologies found in the AI literature. As for the present work, an ontology is defined as system that formally and explicitly describes concepts within the domain of discourse (classes or concepts), whereby the characteristics of each describe its different features and attributes (slots, roles or properties), as well as the limitations on the slots (facets or role restrictions). An ontology plus several individual instances of classes forms a knowledge base. A fine line separates the end of ontologies and the beginning of knowledge bases. The majority of ontologies concentrate on classes, as the latter present a description of the concepts within the domains. Classes may have sub-classes which represent more specific classes as compared to the super-classes. A slot describes the features of the classes and instances [48].

Practically, the development of ontologies requires the following aspects:

- The definition of classes in the ontology,
- The arrangement of the classes in a taxonomic (subclass–superclass) hierarchy,
- The definition of slots and the description of allowed values for these slots,
- The filling-in of values for slots for instances.

Knowledge bases are created through the definition of individual instances of classes which are filled in with particular slot values plus extra slot restrictions [36]. The ontology is a collection of well-defined terms that characterize a specific field. The concepts are described using a hierarchy of subclasses, assigning and specifying properties, and establishing relationships between concepts [49]. Ontologies are the essential elements of the semantic Web. These are documents that describe the relationships of the terms. Ontologies aid with classifying data and information as groups or taxonomies [50].

2.8.1 Ontology representation

Ontology components include four main, popular elements: concepts (classes), instances (individual), relationships, and axioms [51].

• Concepts, also known as classes or categories, are sets of individuals that describe a community of different individuals sharing similar features, which may be more or less unique. The ***SubclassOf (⊆)*** can be used to organize classes into a hierarchical structure or taxonomy relationships. ***A ⊆ B*** refer to A is a ***SubclassOf*** of ***B***. There are at least two classes in all ontologies: the class of all concepts (Thing) and empty set (Nothing), as shown in Figure (2.5). Concepts can also share relationships between each other; these explain the way one concept shares information to another's individuals.

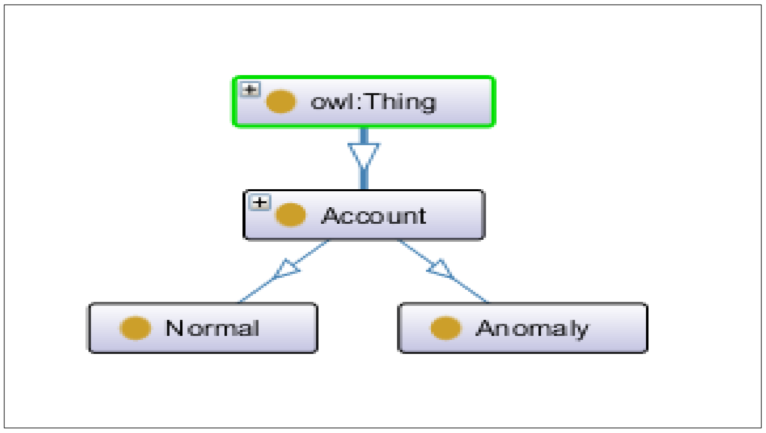


Figure (2.5) Organizing classes into a hierarchical structure or taxonomy relationships.

• Individuals: Individuals (instances) are the cornerstones of ontology, which are the entities that the ontology covers. The most specific concepts expressed in a knowledge base are individual instances [52]. For example, "Human" could be an instance of the class "Animals".

• Relationships: Ontology relationships (Slots or Link) explain how individuals correlate to each other. It is used to convey relationships in a given domain between two concepts the collection of relation types used and their subsumption hierarchy define the ontology's expression power. Ontologies could differentiate between various types of related categories. For instance, between classes, between individuals, between an individual and a class, between a single object and a set, between sets [53].

• **Axioms** the meaning of the axiom is that the individual belongs to the class [54]. denote a statement that is always true, such as "each student is-a person." Axioms are assertions derived from axiomatic statements given in standard order consisting of the overall principle identified by ontology within its implementation domain. Axioms have been used to connect class and property IDs with specifications of their features, either partial or full, and providing other logical class and property information.

2.9 Web Ontology Language (OWL)

The Language of Web Ontology (OWL instead of WOL) is a SW language intended to form rich and intricate knowledge of objects, groups of objects, and relationships among objects. It is an informative definition of language for the development of the ontology. Semantically, OWL was explicitly designed to build web ontologies that resolve the ontological engineering constraints of RDFS with a rich collection of modeling constructors. The first version of OWL began to be developed in 2002, and the second version, OWL 2, began in 2008. In 2004, OWL became a W3C Guideline, and in 2009, OWL 2 was standardized [55].

2.9.1 OWL sublanguages

OWL offers three progressively descriptive sub-languages [56]:

1. **OWL Lite** designed for users who require a hierarchy of classification and simple limitations. OWL Lite offers a rapid migrating path for dictionaries and alternative taxonomies, and it is considered to be relatively simpler than OWL DL.
2. **OWL DL** provides full expressiveness while preserving the completeness and determinability of the computation.
3. **OWL Full** provides full expressive power and the lexical independence of RDF without computation assurances. OWL Full enables the meaning of pre-defined (RDF or OWL) vocabularies to be increased by ontology. For every

function of OWL Full, no reasoning program can actually support a complete reasoning.

2.9.2 OWL Syntax

OWL documents can be presented in various forms of syntax, as illustrated in the following points [57]:

1. **Turtle**: Terse RDF Triple Language (Turtle) is an RDF data model syntax and file format. It is a common data form for storing RDF data. RDF is the knowledge that comprises (subject, predicate, and object) using semantic triple. Each element in the triple format is represented as a Web URI.
2. **RDF / XML**: It is a syntax, identified as the standard RDF format by W3C, which expresses an XML document in the RDF graph. RDF/XML is RDF documents are written in XML language. RDF information can be easily shared among multiple computer types using various platforms and application languages using XML.
3. **Functional syntax**: The OWL functional syntax is the result of the W3C. It is a sequence of Unicode characters accessible from some URI utilizing the standard protocols so that its text matches [57].

Manchester OWL Syntax is a lightweight, human-readable format designed to write OWL class expressions. The CO-ODE project developed Manchester OWL Syntax at the University of Manchester [58].

2.10 The Semantic Web Rule Language (SWRL)

The Semantic Web Rule Language (SWRL) is a W3C recommendation. The OWL / SWRL combination provides a more powerful ontology language with a greater expressiveness for modeling information domains than using OWL on its own. SWRL enables users of writing which are expressed to have more efficient deductive reasoning capabilities as compared to using just OWL. Form a semantic perspective, SWRL is built on the same logical basis as OWL and offers similar emphatic formal assurances when deducing is performed [59]. The Semantic Web expresses the rules of the form (IF condition THEN action). SWRL does not reinforce nonmonotonic assumptions. Thus, the use of SWRL rules will not alter or modify the current data within philosophy. The rules involve an antecedent (e.g., body) and consequent (e.g., head) in the form of reasoning. The conditions mentioned in the precedent must continue in order for the conditions specified in the result to hold. The head and body are composed of one or several atoms joining together, as presented in Equation (2.3) [60].

$$B1 \wedge B2 \wedge \ldots \wedge Bn \rightarrow H \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (\mathbf{2}.3)$$

Where, H: head (an atom) and B1…Bn: body (all atoms), while the atoms in SWRL are defined as follows in Equation (2.4):

$$C(j) \mid O(j,k) \mid D(j,v) \mid :\text{Built-Ins}\ (v1,...,vn) \rightarrow \text{Atom} \quad \ldots\ldots\ldots\ldots\ldots \quad (\mathbf{2}.4)$$

Where C = Class atom such as Account(?a) and Person(Pi)

O = Object Property atom such as has_account(?a,?x)

D = Data type Property

J,k = Object individual or object variable name such as has_src_bytes(?a, ?x) Built-Ins. such as greaterThan () and less Than ()

v1... vn = Data type variable/value name .

The three basic family relationships (***hasParent, isMale, isFemale***) are used to extract new seven relationships (***hasUncle, hasChild, fatherOff, motherOff,***

hasBrother, hasSister, hasSpouse) using SWRL rules in a human readable format, as shown in Table (2.1) [61].

Table (2.1) The SWRL for inferring the new knowledge from existing facts [62].

The basic facts	Description
isMale(?x)	x is a male
IsFemale(?x)	x is a female
hasParent(?x, ?zp)	x has parent p
	<u>Inferring knowledge</u>
hasParent(?x, ?p)	→ hasChild(?p ,?x)
hasParent(?x, ?p)^ hasParent(?x, ?z)	→ hasSpouse(?p, ?z)
isMale(?p)^ hasParent(?x, ?p)	→fatherOff(?p, ?x)
isFemal(?p)^ hasParent(?x, ?p)	→ motherOff(?p ,?x)
isMale(?x)^isMale(?y)^hasParent(?x, ?p)^ hasParent(?y, ?p)	→ hasBrother(?x, ?y)
isFemal(?x)^isFemal(?y)^hasParent(?x, ?p)^ hasParent(?y, ?p)	→ hasSister(?x, ?y)
hasParent(?x, ?p)^ hasBrother(?p, ?z)	→ hasUncle(?x, ?z)

2.11 Semantic Web Development Environments

There are a number of frameworks available which support the OWL ontology. The most commonly used ones are the Jena framework, Protégé-OWL API, and the Wonder Web OWL API, all available for the Java language. Being open-source, the architecture of these three APIs can be studied in-depth. This aspect in particular is of high importance at the present time because of the difficulty in identifying the application scope of the SW in the near future. Open frame-works enable SW components to be integrated easier into new projects.

Protégé is known as a cost-free open-source platform which provides a set of tools for constructing domain models and knowledge-based applications using ontologies [63]. Its initial development was performed by the Stanford Medical Informatics Labs of the Stanford School of Medicine. The Protégé-OWL API is an open-source Java library for OWL and RDF(S). Several classes and methods are provided for loading and storing OWL files, querying and manipulating OWL data models, and for the performance of reasoning (Protégé-API 2006). Extending form the Protégé-OWL plug-in, the Protégé Core System depends on frames in supporting the OWL ontology and allowing users of the development of OWL plug-ins for Protégé or other individual applications. This API makes use of the Jena framework to parse and reason over OWL ontologies, thereby providing extra to program graphical user interfaces (GUIs) based on the Java Swing library. The API architecture takes form of the model-view pattern, which enables users to write GUIs (the "view") for manipulating how ontologies are represented internally (the "model"). This architecture, plus the event mechanisms, enable programmers of building interactive user interfaces efficiently and cleanly [64].

2.12 The Reasoner of Semantic Firewall

A semantic reasoner is software used to create new relationships or information from actual relationships or facts present in an ontology. The objective of reasoning is to extract implicit facts from a collection of explicit facts that are provided. OWL reasoners such as Pellet, HermiT, ELK, and FaCT++ are the most common ontology reasoner in the execution of SWRL rules and the inferring of novel ontology axioms [65]. The reasoning derives information that is not presented directly in ontology or knowledge base, and classification is one of its widest usages. The Pellet reasoner is used in the proposed ontology because of its more direct functionality as it works with OWL and SWRL rules and allowance for defining custom SWRL built-ins.

The Pellet reasoner supports several vital logical services such as consistencies, description, and instance testing, Figure (2.6) shows the main components of the Pellet reasoner. Pellet is the first stable and full OWL-DL reasoner with comprehensive reasoning support for instances, Data types specified by the user, and ontology debugging support [66]. Supposing that x is regarded as an instance, and A, B, C, D are classes. Class A is disjointed from class D. Then, the reasoner can be reasoning the following ontological knowledge [67]:

- **Class Membership**: If $x \in C$ and $C \sqsubseteq D$, then $x \in D$
- **Classes Equivalence**: If $A \equiv B$ and $B \equiv C$ then $A \equiv C$
- **Inconsistency**: Inconsistency means a discrepancy between the axioms that declare individual x belongs to concepts A and D. At the same time, A is disjointed from D, which means that x cannot simultaneously hold in two disjoint concepts.
- **Classification**: The classification of ontology is the measurement of the hierarchies of subsumption for groups and properties automatically. If stating that a property is a necessary prerequisite for a membership, then the assumption is made that an object satisfying this property needs to be an instance of A.

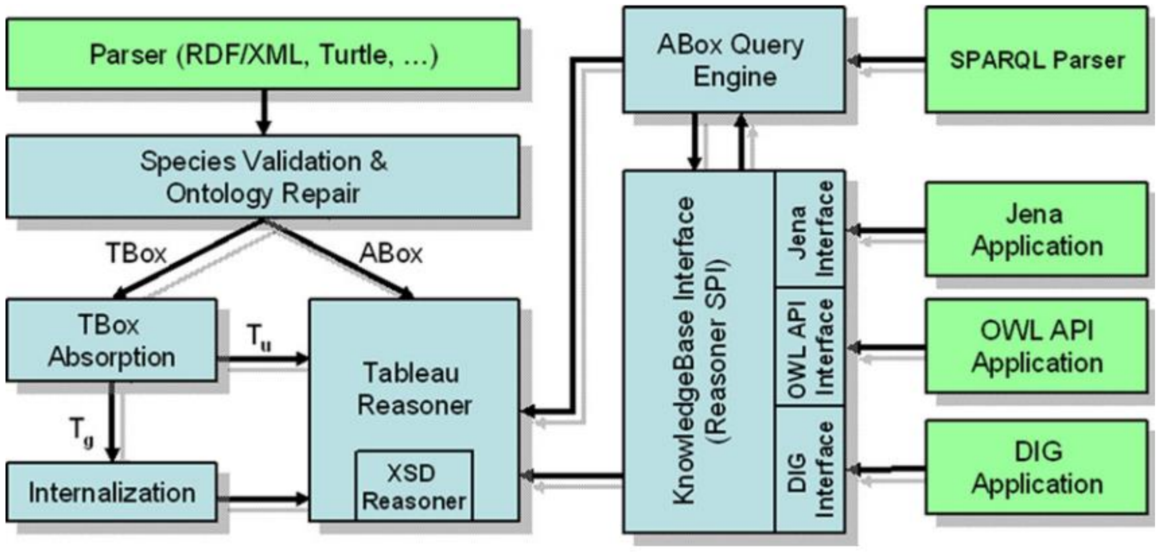


Figure (2.6) Main components of the Pellet reasoner. [55].

The advantages of using ontologies rather than machine learning are listed below:

1) Ontology semantics can be to be both explicit and machine-interpretable (rather than simply machine-processable) [65].
2) An ontology is a human comprehendible, as it is readable and reusable.
3) An ontology represents any forms of data, which can be (un-)structured or semi-structured.
4) An ontology has the fundamental relations among concepts built within them. These allow the data reasoning process to take place automatically.
5) In an ontology, users can navigate more coherently, consistently and easily while moving between concepts.

An ontologies functions similarly to a ‘brain’. It works and reasons with concepts and relations just like humans would perceive interconnected concepts.

2.13 Feature Selection

Feature Selection (FS) is an essential step during the pre-processing stage. Alternative names for FS algorithms include the selection of attributes, instances, data, and variables, as well as feature construction and extraction. FS contributes to the enhancement of data quality as well as the increased accuracy rates of data mining algorithms through the reduction of how complex space and time are. The main focus of FS is the elimination of data that are of high redundancy or irrelevance [27].

2.13.1 Information gain

There are various techniques in the Weka platform to select ML data sets. The (InfoGainAttributeEval) is selected as the most suitable one, as it works on evaluating the values or worth of attributes through the measurement of information gain regarding the classes. This equation (**2**.5) illustrates how to calculate Information Gain [32].

InfoGain (Class, Attribute) = H (Class) – H (Class | Attribute) .... (**2**. 5)

Using the InfoGainAttributeEval for evaluation requires the selection of a search method ranker for ranking the attributes based on the evaluating results. Missing values are treated as separate attribute values. The full data set is used as the training data set. Information Gain is an attributed evaluator adopted in FS whereby the ranker search method is chosen per default. It tends to be biased towards multi-valued attributes. The attribute with the maximum information gain is chosen. Suggesting that (pi) represents the probability of an arbitrary tuple in D belonging to class Ci, as estimated by |Ci, D|/|D|, the expected information (entropy) required for classifying a tuple in D can be stated as follows in equation (**2**. 6) [44].

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2 (p_i) \ldots \ldots\ldots\ldots\ldots \quad (\mathbf{2}.\,6)$$

Information required (after the use of A to split D into v partitions) for classifying D as follows in equation (**2**. 7):

$$Info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times I(D_j) \text{ .... } (\mathbf{2}.7)$$

Information gained through the branching of attribute A can be stated as follows in equation ($\mathbf{2}$. 8):

$$Gain(A) = Info(D) - Info_A(D) \text{... .... } (\mathbf{2}.8)$$

2.13.2 Ranker Search Method

The Ranker method is used in ranking attributes based on the individual evaluations of each. It is often used along with attribute evaluators such as (ReliefF, Gain Ratio, and Entropy). The parameter generates rankings (true/false), a number for selection, and a threshold value for discarding attributes. In general, the ranker method performs the ranks whose attributes ought to be obtaining higher or lower ranks based on the selected attribute within certain data sets. The ranker rates the attributes, and orders them based on their score of evaluators [54].

2.14 Model Evaluation

Several performance measures are used for assessing the consistency of machine learning models. For any model, evaluating machine learning techniques is essential. There are multiple kinds of assessment metrics available for evaluating a model. These include the model's accuracy, Precision, Recall and F-Measure.

2.14.1 Confusion matrix

The confusion matrix is a strategy for outlining classification algorithm results. The percentage of positive and negative predictions is summarized and decomposed by each class by counting values. Confusion matrices are used for assessing the output of classifying algorithms by providing a comprehensive view of how well the identification model works and what sorts of mistakes it makes. All confusion matrix calculation metrics are based on the four basic parameters: True Positives, False Positives, True Negatives, and False Negatives, which are compared in a direct manner. Other classification metrics, such as "Accuracy," on the other hand, provide fewer valuable details, as accuracy is simply the distinction between correct forecasting separated by the overall number of forecasting.

The basic terms metrics of a 2 x 2 matrix for a binary classification query are defined as shown in Figure (2.7) and stated below [68]:

• **True Positive** (TP): Since the real value is positive, and the model's prospection is positive, the predicted value matches the real value.

• **True Negative** (TN): Since the real value is negative, and the model's prospection is negative, the predicted value matches the real value.

• **False Positive** (FP): Since the real value is positive, and the model's prospection is negative, the predicted value mismatches the real value.

• **False Negative** (FN): Since the real value is negative, and the model's prospection is positive, the predicted value mismatches the real value.

Other classification metrics, such as "Accuracy", "ErrorRate", "Precision", "Recall", "F-Measure". The metrics and its formula are explained in Table (2.2).

Table (2.2) Metrics for Classification Evaluations [69].

Metrics	**Formula**	**Evaluation Focus**
Accuracy (acc)	$\frac{TP+TN}{TP+FP+TN+FN}$ .... (**2**. 9)	It is a measurement of the rate of correct predictions to the overall number of instances in evaluation.

ErrorRate (err)	$\frac{\mathrm{FP+FN}}{\mathrm{TP+FP+TN+FN}}$ ……. (**2**. 10)	It measures the fraction of negative patterns which are have a correct classification.
Precision (p)	$\frac{\mathrm{TP}}{\mathrm{TP+FP}}$ …… (**2**. 11)	It measures the positive patterns which have been correctly predicted, out of the overall predicted patterns for a positive class.
Recall (r)	$\frac{\mathrm{TP}}{\mathrm{TP+FN}}$ …… (**2**. 12)	It measures the fraction of positive patterns which have a correct classification.
F-Measure(FM)	$\frac{2*\mathrm{p}*\mathrm{r}}{\mathrm{p+r}}$ …… (**2**. 13)	It presents the harmonic mean shared by the recall and precision values.

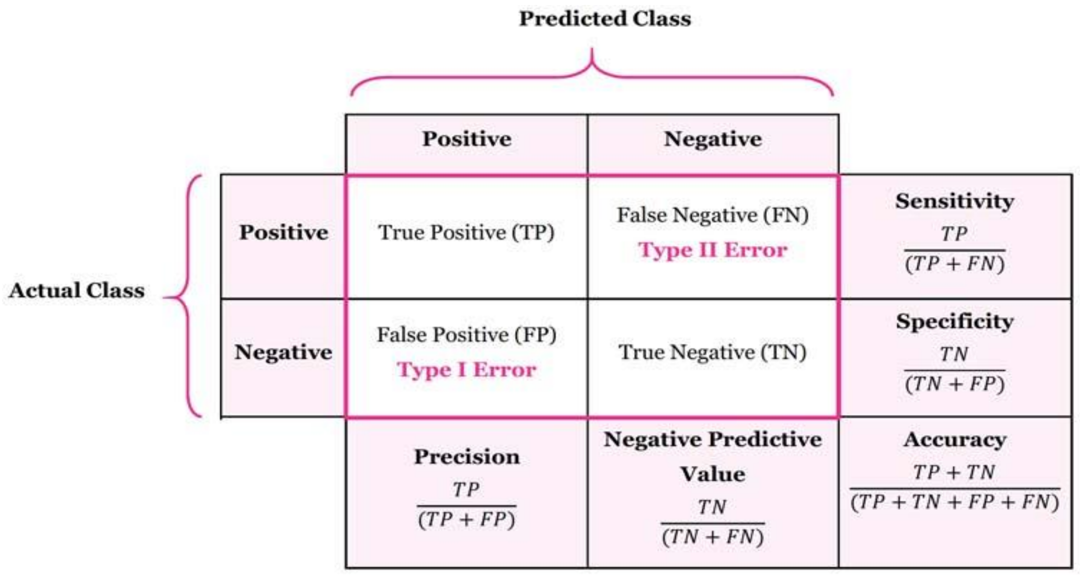

Predicted Class
Positive
Negative
Actual Class
Positive
Negative
True Positive (TP)
False Negative (FN)
Type II Error
False Positive (FP)
Type I Error
True Negative (TN)
Sensitivity $\frac{TP}{(TP+FN)}$
Specificity $\frac{TN}{(TN+FP)}$
Precision $\frac{TP}{(TP+FP)}$
Negative Predictive Value $\frac{TN}{(TN+FN)}$
Accuracy $\frac{TP+TN}{(TP+TN+FP+FN)}$

Figure (2.7) A confusion matrix presents the significant relationships between predicted and actual classes [70].

2.15 The summary

This chapter included the theoretical background about firewalls, the types of machine learning, Semantic Web Technology, Semantic Web Layer Architecture, Uniform Resource Identifier (URI), and UNICODE. It also identifies the ontology, OWL ontology, as well as the OWL and SWRL, explaining the pre-processing methods. In addition, it sheds light on the characterization methods of feature selection as well as the brief explanation of the Reasoner and Confusion metrics.

Chapter Three

The Proposed System and Methodology

3.1 Introduction

This chapter explains the proposed Semantic Firewall of Network intrusion attacks based on machine learning, Description Logic (DL) Reasoners, ontology APIs, and semantic web languages (OWL and SWRL). This is followed by a description of the data sets used in Network Intrusion Detection. Finally, it explains all implementation steps of this thesis.

3.2 The Proposed System description

This system presents a proposal to detect network intrusion attacks, which affect network security. For detecting this type of attack, machine learning, and OWL ontology are used. Figure (3.1) illustrates the main stages of which the proposed system is composed. The first stage is the data pre-processing and selection of features through machine learning techniques The data in its current form may not be suitable for predictive processing, and therefore needs pre-processing. Data pre-processing is the preparatory process of deleting duplicated, unnecessary, incorrectly formatted, missing data, and data type conversion, whereby raw data is transformed into a more useful format. The second stage, the decision tree technique is an inductive (ML) technique used for building decision tree rules to get the threshold. A threshold value determines the previous probabilities could be interpreted as class labels to be classified as binary scoring. The threshold is the range of criteria values that determine the positive conditions that provide the maximal rate of accuracy The third stage, ontology engineering is utilized in the creation of ontologies and knowledge representations. These are used in decision tree rules through their conversion into rules-based reasoners of SWRL that are used in the detection of anomalies of network attacks.

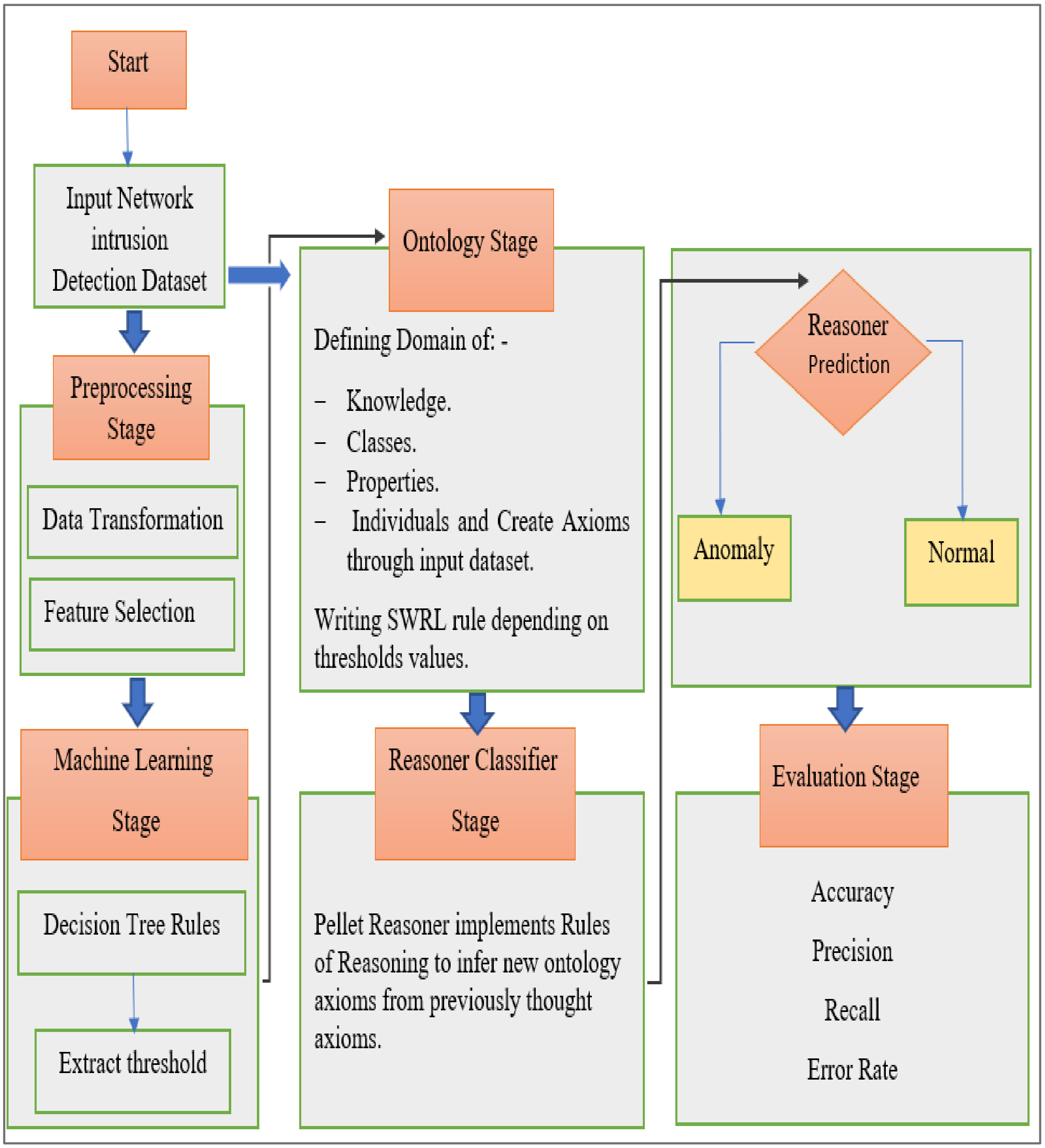

Start
Input Network intrusion Detection Dataset
Preprocessing Stage
Data Transformation
Feature Selection
Machine Learning Stage
Decision Tree Rules
Extract threshold
Ontology Stage
Defining Domain of: -
– Knowledge.
– Classes.
– Properties.
– Individuals and Create Axioms through input dataset.
Writing SWRL rule depending on thresholds values.
Reasoner Classifier Stage
Pellet Reasoner implements Rules of Reasoning to infer new ontology axioms from previously thought axioms.
Reasoner Prediction
Anomaly
Normal
Evaluation Stage
Accuracy
Precision
Recall
Error Rate

Figure (3.1) Block diagram steps of the proposed system.

3.3 Dataset Description

This section presents some background information about the Network Intrusion Data set that is used throughout the present work. All connection records are around 100 bytes long. Out of the normal and attack data, 41 quantitative and qualitative characteristics are extracted for every TCP/IP connection (38 quantitative features and 3 qualitative). The dataset consisted of (25192) rows and (42) columns The class variable has two categories:

- Normal
- Anomalous

3.3.1 Data Pre-processing and Features Selection

The pre-processing step must be implemented before any process, as it manipulates the data into a form that machine learning can deal with. After pre-processing, the data set consists of 7650 records and 13 features. The pre-processing stage involves feature dropping (Deprecated features, too many missing values, Blank fields, and Irrelevant features) and types conversion.

A.Features dropping

The dataset may include irrelevant, incomplete, and blank data fields; such fields should be dropped from the dataset. The dropping fields are illustrated as follow:

1. **Deprecated features:** The dataset includes meaningless information. It was linked to attributes that have never been revealed and are now deprecated, such as (protocol_type) and (diff_srv_rate).
2. **Fill in many missing values:** The dataset consists of several fields with too many missing values, such as is_guest_login, num_outbound_cmds, root_shell, and su_attempted.
3. **Blank fields:** The dataset contains seven fields with no data such as srv_count, srv_diff_host_rate, and dst_host_count.

4. **Irrelevant features:** Some features are not used to classify tasks, such as dst_host_serror_rate, rerror_rate, srv_serror_rate.

B. Type conversion

The dataset includes some value types of features that need to be converted into an integer type. Some features need to be coded and converted into integer values such as service and flag by sorting. For example, in the case of the service feature, the services are sorted alphabetically in ascending mode, after which each service in the sorted list is coded from 1 to end of list 64 in excel sheets.

3.3.2 Feature Selection

Feature selection (FS) is considered to be among the principal concepts in machine learning that significantly affect the model's performance. The process involves choosing the minimum necessary features to create a reasonable model. Given that the more features a model includes, the more complicated it is, therefore it would be more susceptible to variance errors. The selection of important features can be achieved using the vital scores for selecting specific attributes to be eliminated (lowest ratings) or those features to be kept (highest ratings). The significant features have several advantages that include simplifying the issue being developed, speeding up the modeling process, and increasing productivity in some cases. The advantages of selecting important features can also lead to better data understanding, model understanding, and reduced input features. The selecting of attributes is a process that consists of two steps, namely generating sub-sets and rankings. The first represents the searching process for comparing candidate sub-sets to the predetermined sub-sets. In case the candidate sub-set provides better results, then the new sub-set is considered to be the best. This procedure is continued until reaching the termination condition. The second step involves assigning ranks to attributes based on how important they are found to be. Inappropriate features found within the data lead to the reduction of how accurate the model is, eventually causing it to perform the learning processes according to features that have no relevance. Therefore, the features were chosen according to the ones that are most affected by Network intrusion attacks.

3.3.2 .1 Information Gain Attribute of Dataset (IG):

The method aims to measure how important attributes are based on the IG filter whose calculation is made in light of the target class. The following equation illustrates how its values can be obtained:

$$\text{InfoGain (Class, Attribute)} = H(\text{Class}) - H(\text{Class} \mid \text{Attribute})$$

Where H is the Entropy. This attribute evaluator is combined with the Ranking method of Search in its application through the Network intrusion data set. The ranks of the first twelve attributes are taken into records, as follows in Table (3.1):

Table (3.1) Final features after using Attribute selection techniques

No.	Features name
1	service
2	flag
3	src_bytes
4	dst_bytes
5	logged_in
6	count
7	serror_rate
8	dst_host_srv_count
9	dst_host_same_srv_rate
10	dst_host_diff_srv_rate

11	dst_host_srv_diff_host_rate
12	dst_host_srv_serror_rate

3.4 Technologies of the proposed system

As for the present thesis, there are two types of technology implemented, namely ML and SW. Being the first to invent the internet and define SW, Berners-Lee aimed towards indexing the Internet using semantics rather than keywords, through the use of ontologies so that the sets, sub-sets, relations, and rules for various domains can be described. They underpin the SW, which directly enables machines of processing data. The OWL and SWRL are the W3C standards for the definition of ontology and rules, respectively.

3.4.1 Machine Learning Technologies

Machine learning refers to the concept that involves learning from experience (previous data) for improving future performances. The algorithms of Machine learning build a model based on a sample of data, known as "training data", to make predictions or decisions without being explicitly programmed to do so.

3.4.1.1 J48 Decision Tree Algorithm for Calculating Threshold

J48 one of the commonly used classifying techniques is the Decision tree, as it depends on dividing and conquering strategies. DT was used in proposed than others because it gave high accuracy of extract features threshold. Decision trees consist of decision and leaf nodes. The first type specifies a test over one of the attributes, whereas the second represents the class values. The paths from the root to leaf nodes are called rules, as shown in Figure (3.2) Algorithm (1). The initial state of a decision tree is the root node that is assigned to all the examples from the training set, as shown in Table (3.2) the steps of Decision tree Construction. Therefore, the rule that results from the decision tree algorithm has a threshold value, and these values represent the fundamental of writing the SWRL rule. A threshold value is decided by J48. The accuracy of the decision tree for getting the threshold is 99%, so written rules in SWRL are based on the best threshold.

Figure (3.2) Algorithm 1: Pseudocode for (J48) algorithm for calculating Threshold.

```
     INPUT: N, where N = Root node
     OUTPUT: Decision tree rule
1:       If (T belongs to same category C)
2:           {leaf node=N;
             mark N as class C;
             return N;}
3:      For i =1 to n
            {Calculate Information_gain (A$i$);}
4:           $t_a$ = testing attribute;
5:          N. $t_a$ = attribute having highest information gain;
6:        if (N. $t_a$ == continuous)
           {find threshold;}
7:     For (Each T’ in the splitting of T)
8:        if (T’ is empty)
             {child of N is a leaf node;}
9:          else
              {child of N= dtree (T’)}
10:           calculate classification error rate of node N;
11:        return N;
12:      end if
13:   end for
14: end if
```

Table (3.2) Steps of Decision tree Construction.

```
Step 1.  All cases need to be checked whether they belong to the same class,
    if
        so,
    then
      both the tree and leaf will be assigned the same class label.

Step 2.  The information and information gain is calculated for all attributes.

Step3. The best splitting attribute is obtained based on the present selecting
    criteria.
End
```

3.5 Ontology Construction

The OWL ontology design declares the domain terminologies and relationships appearing as OWL entities (owl:Class; owl:ObjectProperty; owl:DatatypeProperty). They are labeled with a readable form label (rdfs:label) and linked with a clear, succinct, and precise description (rdfs:comment). The Protege tool are taken into consideration because it has a user-friendly interface for building and modifying ontologies, such as ontology metrics as well as extra plugins for visualizing them, such as NavigOWL. The fundamental elements of the proposed ontology are as shown in Table (3.3) Steps of Ontology Construction.

Table (3.3) Steps of Ontology Construction

Step 1. Determine the domain and scope of the ontology

/ For what we are going to use the ontology? */*

Step 2. Define the classes and the class hierarchy

/ a class is a collection of individuals or objects. */*

Step3. Arranging the classes in a taxonomic (subclass –superclass) hierarchy

Step 4. Define the properties of classes

/ Object properties link individuals to individuals. */*

/ Datatype properties link individuals to data values. */*

Step 5. defining slots and describing allowed values for these slots,

/ Slot cardinality defines how many values a slot can have. */*

*/*Slot-value type describes what types of values can fill in the slot. String, Number, Boolean, Enumerated */*

Step 6. Create instances

/ Creating individual instances of classes */*

The elements of an ontology developed on OWL are summarized as follow:

a. Individuals: individuals are the instances of a class.

b. Classes: classes reflect the definitions in a definite way.

c. Properties (slots): object and class characteristics, also known as descriptive logic roles.

d. Relations: the forms through which classes and objects are linked.

e. Terms of function: they are complicated structures created by specific relationships that can be used rather than an individual member of a declaration.

f. Restrictions: a formalized definition of what must be valid for that to be recognized as an entry in a statement given.

g. Rules: if-then expressions, defining the logical inference that can be derived in a specific form from a statement.

h. Axioms: statements (including rules) in the logical form grouped encompass all the theories defined in the field of application by ontology.

When creating knowledge bases, the Stanford Protege ontology editor is used for establishing OWL ontologies in representing the model classes and features. There are two main classes involved, namely the accounts and the persons. Figure (3.2) illustrates the class hierarchy in the semantic firewall ontologies, according to the visualization provided by the OntoGraf ontology visualization tool. The OntoGraf is a plug-in tab in protege's editor used to visualize and interactively navigate the relationships of our OWL ontology. Various layouts are supported for automatically organizing the structure of the ontology. Different relationships are supported: subclass, individual, domain/range object properties, and equivalence.

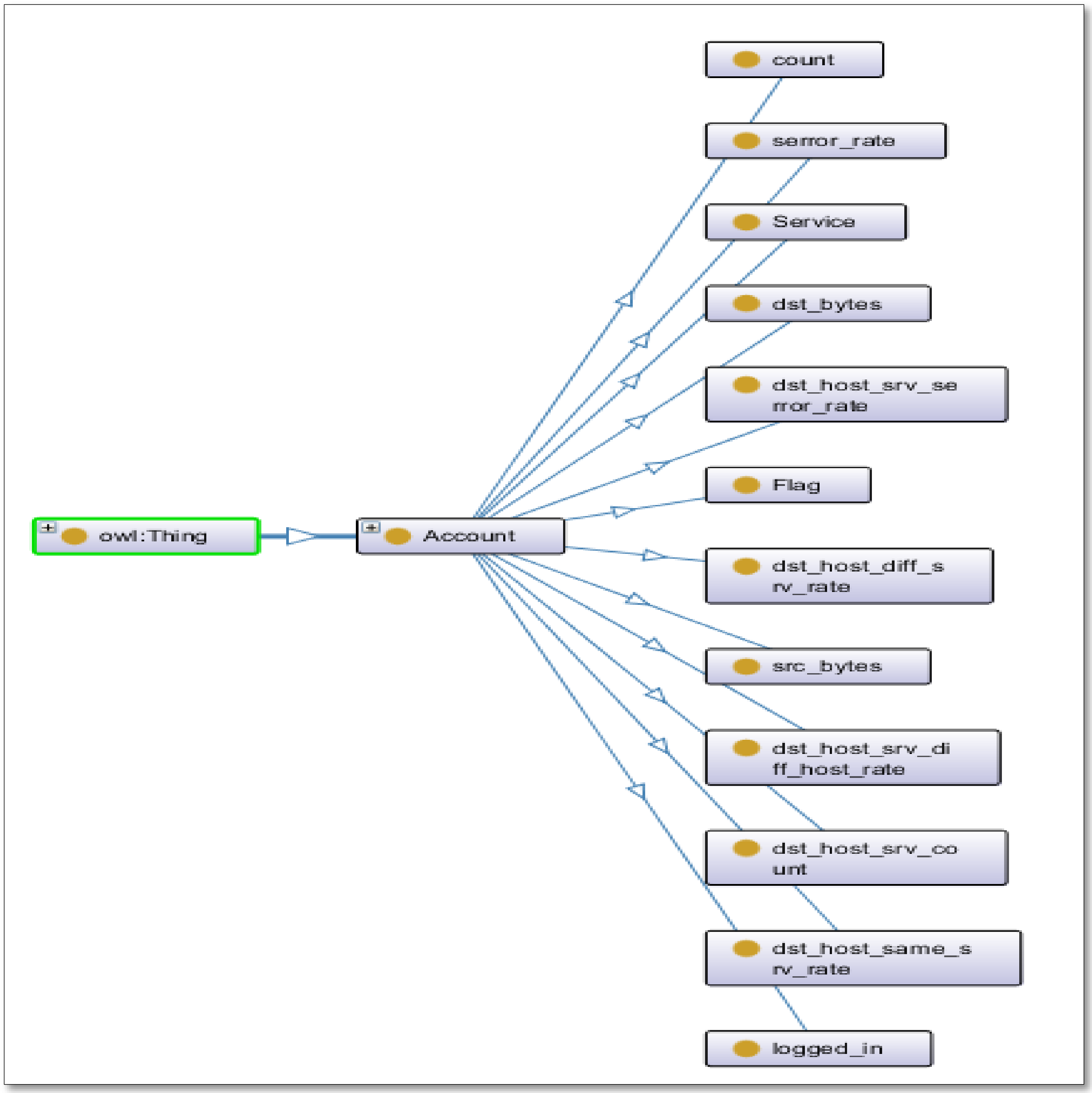


Figure (3.2) The tree hierarchy of account class is visualized in protégé.

3.5.1 Classes of Semantic firewall (SFW) ontology

There are several ways a class hierarchy is constructed. The top to bottom approach begins by defining everything that holds the most general possible class a group of meanings might accept. It successively redefines each subset separately. Using groups, the general concepts can be described such as accounts. Next, some specialized account features are built. The main classes of SFW ontology are: the Person class and the Account class. In class, hierarchy means: class X is a subclass of Y such as the Middle level as shown in Figure (3.3). In contrast, the Reliable class and confident class are not disjoint, because they share the same concepts.

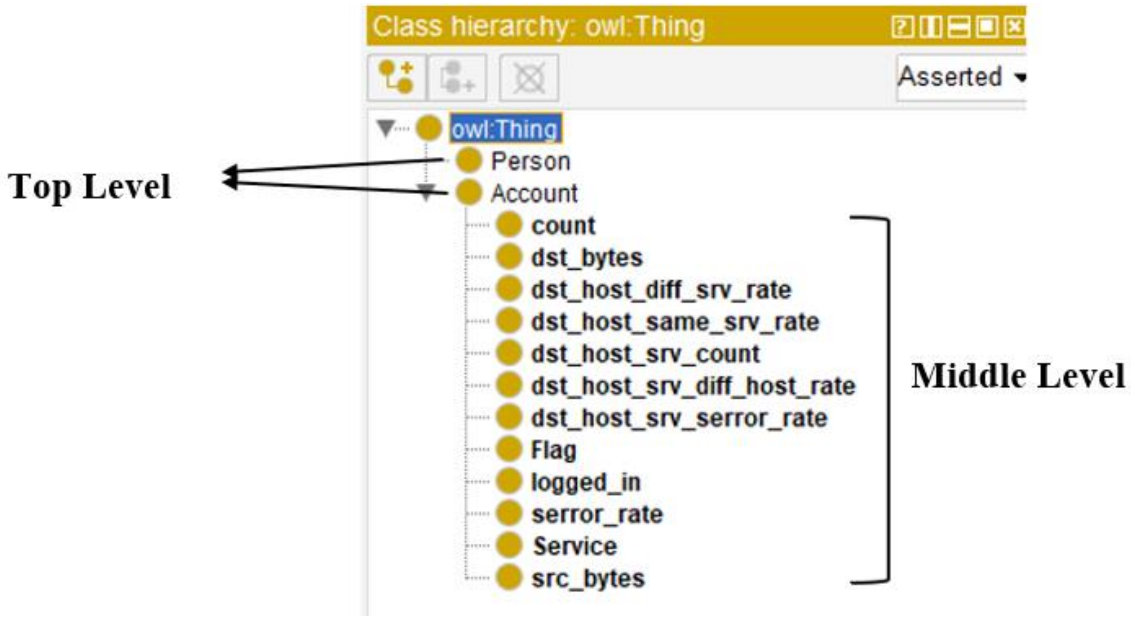


Figure (3.3) A top-down class hierarchy approach

3.5.2 Classes Properties (Slots)

Developing classes that have properties is a prerequisite for building an ontology. Classes by themselves are not adequate in supplying information. Object properties are essential in the proper classification and categorization of ontology. Inheritance within ontology classes is related to the inheritance of properties from parent classes. In the OWL ontology, objects are categorized into two distinct categories: data properties and entity properties. Data properties are binary relationships that join an individual to data value type such as "xsd: integer" The object's properties are binary relationships connecting an entity to another entity.

3.5.3 Slots Facets

The slots have various facets that determine the form of value, the permitted value, the number of cardinal values, and alternative value characteristics that could be taken up by the slot. The common facets include:

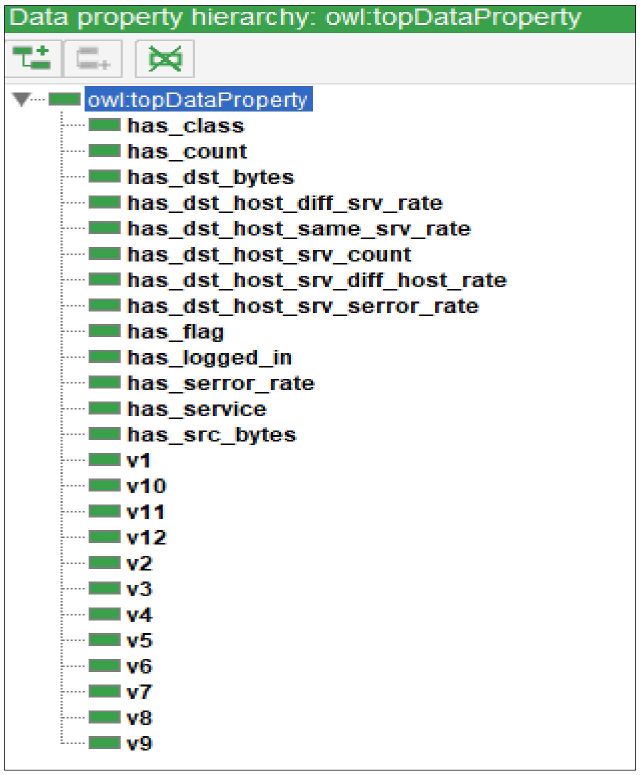
• The slot cardinality (the number of slots) specifies values for a given slot. Some schemes maintain only one type of cardinality, while others preserve several types (enabling any range of elements).

• The slot specifies the permissible values a property can contain, a lot of which are strings, integers, and Boolean. Table (3.4) and Figure (3.4) list the data properties of the SFW ontology.

Table (3.4) Data properties of SFW ontology.

Data Property	Domain	Rang
has_class	Account	xsd:decimal
has_count	Account	xsd:decimal
has_dst_bytes	Account	xsd:decimal
has_dst_host_diff_srv_rate	Account	xsd:decimal
has_dst_host_same_srv_rate	Account	xsd:decimal
has_dst_host_srv_count	Account	xsd:decimal
has_dst_host_srv_diff_host_rate	Account	xsd:decimal
has_dst_host_srv_serror_rate	Account	xsd:decimal
has_flag	Account	xsd:decimal
has_logged_in	Account	xsd:decimal
has_serror_rate	Account	xsd:decimal
has_service	Account	xsd:decimal
has_src_bytes	Account	xsd:decimal
v1	Account	xsd:decimal

v10	Account	xsd:decimal
v11	Account	xsd:decimal
v12	Account	xsd:decimal
v2	Account	xsd:decimal
v3	Account	xsd:decimal
v4	Account	xsd:decimal
v5	Account	xsd:decimal
v6	Account	xsd:decimal
v7	Account	xsd:decimal
v8	Account	xsd:decimal
v9	Account	xsd:decimal


Data property hierarchy: owl:topDataProperty

```
owl:topDataProperty
    has_class
    has_count
    has_dst_bytes
    has_dst_host_diff_srv_rate
    has_dst_host_same_srv_rate
    has_dst_host_srv_count
    has_dst_host_srv_diff_host_rate
    has_dst_host_srv_serror_rate
    has_flag
    has_logged_in
    has_serror_rate
    has_service
    has_src_bytes
    v1
    v10
    v11
    v12
    v2
    v3
    v4
    v5
    v6
    v7
    v8
    v9
```

Figure (3.4) Data properties of SFW ontology as visualized in Protégé editor.

3.6 SWRL Rules Creation

The If-then statement is the most fundamental declaration of decision making. It is used to determine whether or not a particular assertion or block of statements is executed, i.e., when a particular assumption is valid, a block of statements is done unless not. A conditional rule (also called an If-Then Statement) is represented by amplification formula B→H. The amplification formula is an if-then expression in which B is an assumption and H an inference. The symbol → denotes the logical link in a conditional statement. The conditional is defined to be true unless a true condition drives a false conclusion.

In short, the SWRL rule has an IF-THEN format, and its head and body consist of atoms that evaluate components in the existing knowledge and perform simple calculations using SWRL's built-in function collection. Atoms in the body may be regarded as skillset data queries, while atoms in the head reflect assumptions that change the knowledge base. The additional layer of expressivity can be provided by adding rules to OWL using SWRL. SWRL rules are used to derive new information from present knowledge. In terms of ontological concepts, all the rules are expressed as (classes, properties, and individuals).

Identifying hypotheses is the cornerstone of any decision rules needing to determine the threshold. Threshold (the cutoff point) is used in hypotheses to formulate a suitable condition. In other words, a threshold represents the separating point between fulfilling the condition or not.

The selection of significant related features determines the important rules that will bring about the better outcome. The rules are translated into SWRL and Pellet reasoner is then applied to infer whether the record is an anomaly or normal. The final classification decision depends on the final aggregated score. The count estimation depends on point-scoring, which is a cumulative summation outcome of all rules. Each rule gains zero points or one point according to specific predefined criteria. For example, on a binary classification problem with class labels 0 (normal) and 1 (anomaly), a normalized predicted threshold of (8 for feature service). This implies assigning any values that are less than or equal to the threshold of 6 to class 0, and any value greater than 6 to class 1 as shown in Table (3.5), Table (3.6) shows the Rules for calculating evaluation metric. The Final Rule is to gather all thresholds that determine whether the record is an anomaly (score is greater than 6) or normal (score is less than or equal to 6), as shown in Formula (1), (2).

- Prediction <= 5 ------------ Class 0 ……….. (1)
- Prediction > 5 ------------ Class 1 ……….. (2)

Table (3.5) SWRL rules of SFW Ontology.

Rule greater Than	**Rule lessThanOrEqual**
Account (? a) ^ has_service(?a, ?x) ^ swrlb:greaterThan(?x,8) → v1(? a, 1)	*Account (? a) ^ has_service(? a,?x) ^ swrlb:lessThanOrEqual(?x, 8) → v1(? a, 0)*
Account (? a) ^ has_flag(?a, ?x)^ swrlb:greaterThan(?x, 5.5) → v2(? a, 1)	*Account (? a) ^ has_flag(?a,?x)^ swrlb:lessThanOrEqual(? x, 5.5) →v2(? a, 0)*
Account (? a) ^ has_src_bytes(?a,?x)^ swrlb:greaterThan(?x, 28.5) →v3(?a, 0)	*Account (? a) ^ has_src_bytes(?a, ?x) ^ swrlb:lessThanOrEqual(?x, 28.5) →v3(?a, 1)*
*Account (? a) ^ has_dst_bytes(?a,?x)^*	*Account (? a) ^ has_dst_bytes(?a,?x)^*

swrlb:greaterThan(?x, 1) → v4(?a, 0)	*swrlb:lessThanOrEqual(?x, 1) → v4(?a, 1)*
Account (? a) ^ has_logged_in(?a,?x)^ swrlb:greaterThan(?x,0) → v5(?a, 0)	*Account (? a) ^ has_logged_in(?a,?x)^ swrlb:lessThanOrEqual(?x, 0) → v5(?a, 1)*
Account(?a)^ has_count(?a,?x)^ swrlb:greaterThan(?x,2) → v6(?a, 1)	*Account (? a) ^ has_count(?a,?x) ^ swrlb:lessThanOrEqual(?x, 2) →v3(? a, 0)*
Account(?a)^ has_serror_rate(?a,?x)^ swrlb:greaterThan(?x,0.73) → v7(?a, 0)	*Account(?a) ^ has_serror_rate(?a,?x)^ swrlb:lessThanOrEqual(?x, 0.73) →v7(?a, 1)*
Account(?a) ^ has_dst_host_srv_count(?a,?x)^ swrlb:greaterThan(?x,89) →v8(?a, 0)	*Account(?a)^ has_dst_host_srv_count(?a,?x)^ swrlb:lessThanOrEqual(?x, 89) →v8(?a, 1)*
Account(?a)^ has_dst_host_same_srv_rate(?a,?x)^ swrlb:greaterThan(?x, 0.53) → v9(?a, 0)	*Account(?a)^ has_dst_host_same_srv_rate(?a,?x)^ swrlb:lessThanOrEqual(?x, 0.53) → v9(?a, 1)*
Account(?a)^ has_dst_host_diff_srv_rate(?a,?x)^ swrlb:greaterThan(?x, 0.89) → v10(?a, 1)	*Account(?a)^ has_dst_host_diff_srv_rate(?a,?x)^ swrlb:lessThanOrEqual(?x, 0.89) → v10(?a, 0)*

Account(?a)^ has_dst_host_srv_diff_host_rate(?a, ?x) ^ swrlb:greaterThan(?x,0.25) →v11(?a, 1)	*Account(?a)^ has_dst_host_srv_diff_host_rate(?a, ?x) ^ swrlb:lessThanOrEqual(?x, 0.25) →v11(?a, 0)*
Account(?a)^ has_dst_host_srv_serror_rate(?a,?x)^ swrlb:greaterThan(?x, 0.79) → v12(?a, 1)	*Account(?a)^ has_dst_host_srv_serror_rate(?a,?x)^ swrlb:lessThanOrEqual(?x, 0.79) → v12(?a, 0)*
Account(?a) ^ v1(?a,?x1)^ v2(?a, ?x2) ^ v3(?a, ?x3) ^ v4(?a, ?x4) ^ v5(?a, ?x5) ^ v6(?a, ?x6) ^ v7(?a, ?x7) ^ v8(?a, ?x8) ^ v9(?a, ?x9) ^ v10(?a, ?x10) ^ v11(?a, ?x11) ^ v12(?a, ?x12) ^ swrlb:add(?sum, ?x1, ?x2, ?x3, ?x4, ?x5, ?x6, ?x7, ?x8, ?x9, ?x10, ?x11, ?x12) ^ swrlb:greaterThan(?sum,6) →Anomaly(?a)	*Account(?a) ^ v1(?a,?x1)^ v2(?a, ?x2) ^ v3(?a, ?x3) ^ v4(?a, ?x4) ^ v5(?a, ?x5) ^ v6(?a, ?x6) ^ v7(?a, ?x7) ^ v8(?a, ?x8) ^ v9(?a, ?x9) ^ v10(?a, ?x10) ^ v11(?a, ?x11) ^ v12(?a, ?x12) ^ swrlb:add(?sum, ?x1, ?x2, ?x3, ?x4, ?x5, ?x6, ?x7, ?x8, ?x9, ?x10, ?x11, ?x12) ^ swrlb:lessThanOrEqual(?sum, 6) →Normal(?a)*

Table (3.6) Rules for calculating evaluation metrics.

Normal (?a) ^ has_class(?a, ?c) ^ swrlb: equal (?c, 1) →FP(?a)	*Normal(?a) ^ has_class(?a, ?c) ^ swrlb:equal(?c, 0) →TP(?a)*
Anomaly(?a) ^ has_class(?a, ?c) ^ swrlb:equal(?c, 0) →FN(?a)	*Anomaly(?a) ^ has_class(?a, ?c) ^ swrlb:equal(?c, 1) →TN(?a)*

3.7 The Reasoner of SFW ontology approach

A reasoner is a software application, used to derive implied facts through several explicit facts. The expression of the latter may take the form of ontologies and are stored RDF triple stores. One of the most widely known usages of the reasoner is classification. The reasoning tasks in OWL involve how consistent the ontologies are, how the classes can be satisfied, as well as the instance checking and conjunctive query answering. A semantic reasoner derives logical consequences from a collection of stated axioms in an ontology and often supports reasoning activities such as classification and querying automatically. A semantic reasoner's judgments about an ontology generated by the Concept Modeler can be utilized to identify logical flaws in the core concept model. As a result, a semantic reasoner can provide information that can be used to validate and enhance a concept model. The OWL reasoner is a method that can infer logical conclusions from a set of asserted facts. Reasoner implements Rules of Reasoning to infer new ontology axioms from previously thought axioms. Pellet is a full OWL reasoner with a good performance that enables users to conduct a hierarchical classification of resources and assess ontologies' consistency. The consistency is checked by analyzing the coherence of a series of hypotheses verifying that no inconsistencies are present.

3.8 Summary

This chapter illustrated the proposed model, which included the input dataset, project attributes, and the division of the processed data (training and testing) to be used by the chosen algorithms to the Network Intrusion attacks. After evaluating the performance to ensure the model's validity and efficiency, the protection of the training data is explained. Ontology is a model for the representation of information that can be shared and re-used around a domain. Their ability to explain relationships and high interconnectivity create the foundations for high-quality, connected, and coherent data modeling. The architecture of ontology steps is determining an ontology domain and range, defining the classes and class hierarchy, defining class properties, determining facts of the slots, writing SWRL rules, and finally invoking a reasoner to identify the anomaly and classify it. The next chapter implements and evaluates these algorithms to obtain the results.

Chapter Four

Implementation and results

4.1 Introduction

This chapter presents a new approach to Detect a firewall anomaly using ontology engineering and semantic web rules. The used dataset of Network Intrusion Detection consists of 7649 records and 13 features, with a testing set of 7649 records, distributed into 4557 normal items and 3092 anomalies. The dataset is split into two sets of training (70%) and testing (30%). The model scenario is the use of an ontology, along with semantic rules and reasoners for learning the rules which classify anomalies. In addition, an ontology classifier is used as a decipherable model with simple and human-interpretable decision rules.

4.2 Dataset description

The data set provides a varying range of intrusions that are mimicked within military network environments, given for auditing. By replicating a typical US Air Force LAN, it provided a context in which raw TCP/IP dump data of networks could be obtained. The total Number of Instances are 25192, with Normal (53%) and Anomaly (47%) and 42 features as shown in Table (4.1). The class variable has two categories: Normal and Anomalous.

Table (4.1) The dataset features.

No.	**Feature name**
1.	duration
2.	protocol_type
3.	service
4.	flag
5.	src_bytes
6.	dst_bytes

7.	land
8.	wrong_fragment
9.	urgent
10.	hot
11.	num_failed_logins
12.	logged_in
13.	num_compromised
14.	root_shell
15.	su_attempted
16.	num_root
17.	num_file_creations
18.	num_shells
19.	num_access_files
20.	num_outbound_cmds
21.	is_host_login
22.	is_guest_login
23.	count
24.	srv_count
25.	serror_rate
26.	srv_serror_rate
27.	rerror_rate
28.	srv_rerror_rate
29.	same_srv_rate
30.	diff_srv_rate
31.	srv_diff_host_rate
32.	dst_host_count

33.	dst_host_srv_count
34.	dst_host_same_srv_rate
35.	dst_host_diff_srv_rate
36.	dst_host_same_src_port_rate
37.	dst_host_srv_diff_host_rate
38.	dst_host_serror_rate
39.	dst_host_srv_serror_rate
40.	dst_host_rerror_rate
41.	dst_host_srv_rerror_rate
42.	class

4.3 Data Preprocessing

The data in its current form may not be suitable for predictive processing, and therefore needs pre-processing. Data pre-processing is the preparatory process of deleting duplicated, unnecessary, incorrectly formatted, missing data, and data type conversion, as shown in Table (4.2).

Table (4.2) List of dropdown features.

No.	Feature name	status
1.	duration	too many missing values
2.	protocol_type	Deprecated
3.	land	too many missing values
4.	wrong_fragment	too many missing values
5.	urgent	too many missing values
6.	hot	too many missing values

7.	num_failed_logins	too many missing values
8.	num_compromised	too many missing values
9.	root_shell	too many missing values
10.	su_attempted	too many missing values
11.	num_root	too many missing values
12.	num_file_creations	too many missing values
13.	num_shells	too many missing values
14.	num_access_files	too many missing values
15.	num_outbound_cmds	too many missing values
16.	is_host_login	too many missing values
17.	is_guest_login	too many missing values
18.	srv_count	Blank field
19.	srv_serror_rate	irrelevant
20.	rerror_rate	irrelevant
21.	srv_rerror_rate	irrelevant
22.	same_srv_rate	Blank field
23.	diff_srv_rate	Deprecated
24.	srv_diff_host_rate	Blank field
25.	dst_host_count	Blank field

26.	dst_host_same_src_port_rate	Blank field
27.	dst_host_serror_rate	irrelevant
28.	dst_host_rerror_rate	Deprecated
29.	dst_host_srv_rerror_rate	Deprecated

The data set includes several value types of features that need to be converted into an integer type, whereby all string types have been converted to integer type. Table (4.3) shows the final essential features.

Table (4.3) The cleaned dataset features

Feature name	**Type**
Service	String
Flag	String
src_bytes	Int
dst_bytes	Int
logged_in	Int
Count	Int
serror_rate	Int
dst_host_srv_count	Int
dst_host_same_srv_rate	Int
dst_host_diff_srv_rate	Int
dst_host_srv_diff_host_rate	Int
dst_host_srv_serror_rate	Int

4.4 Experimental Environment

The environment characteristics that are used is present in Table (4.4), with Python as the programming language, and protégé for semantic technologies. The simplicity of Python permits promoters to write dependable systems. In addition, Python calls a lot of promoters because it is simple to learn. The Python code can be understood by users, thereby making it simple to structure algorithms for ML. The Protege-OWL API can be considered as an open-source Java library for OWL and RDF(S).

Table (4.4) Environment characteristics.

OS	**Windows 10 pro**
Processor	Core (TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
RAM	20.0 GB
Language	Weka, Python 3.8, Protégé 5.5.0

4.5 Feature Selection

In the filter feature-selection approach, the modified data set has been successfully implemented after selecting the attributes, particularly with respect to the InfoGainAttributeEval and Ranker Search Method, as shown in Table (4.5), and Figure (4.1) ranked attributes according to the results of the feature ranking techniques, all twelve features are arranged in descending order from the highest value to the lowest value ranks. After Preparation dataset using selection techniques Figure (4.2) shows the Dataset after preparation.

Table (4.5) Feature ranking using Attribute selection techniques.

Selected Attributes	**Ranked Attributes**
src_bytes	0.797
Service	0.677
dst_bytes	0.628
Flag	0.523
dst_host_srv_count	0.459
dst_host_same_srv_rate	0.431
dst_host_diff_srv_rate	0.413
logged_in	0,412
dst_host_srv_serror_rate	0.405
serror_rate	0.395
count	0.383
dst_host_srv_diff_host_rate	0.267

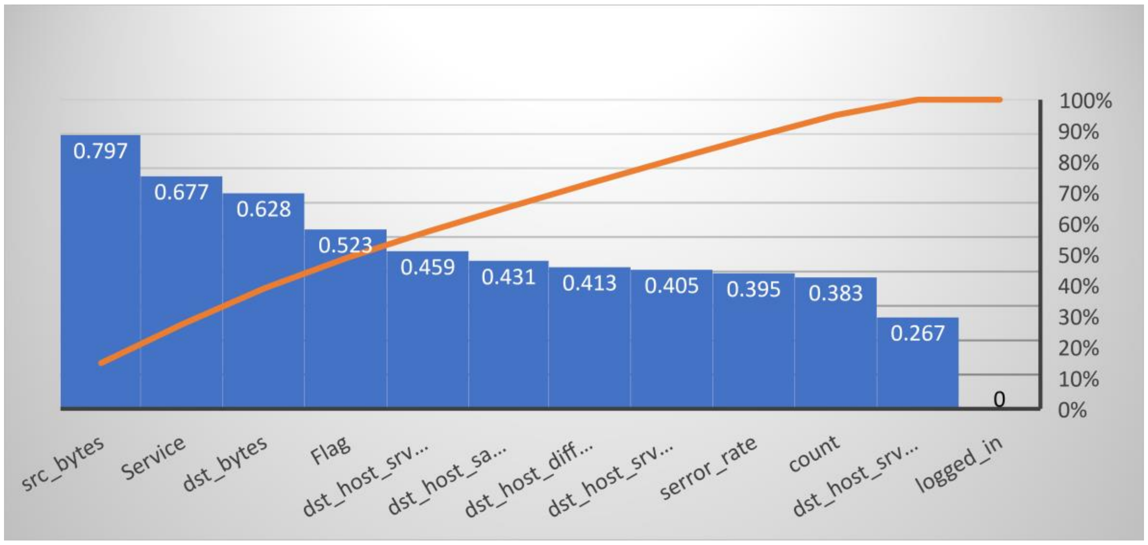
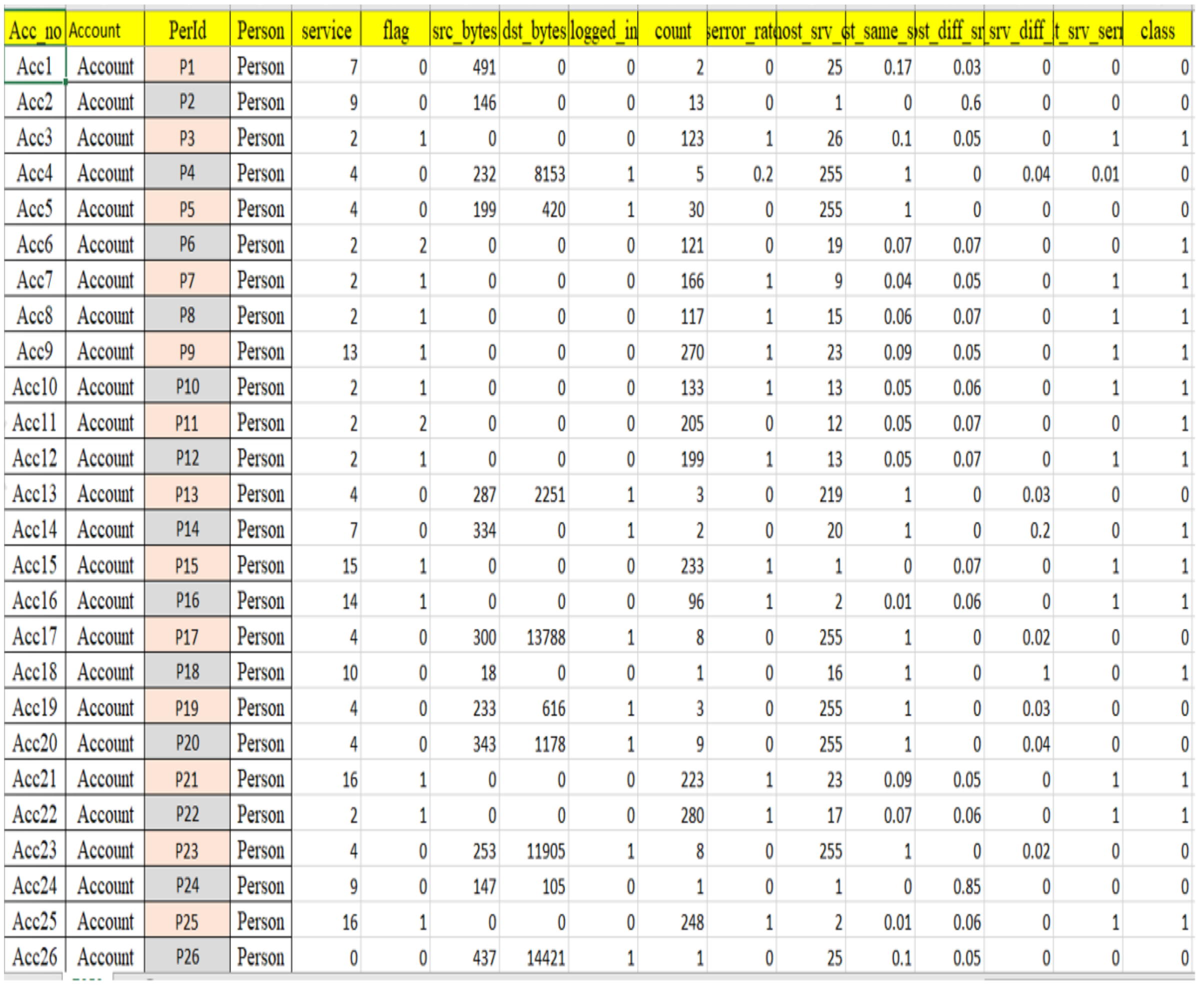


Figure (4.1) Ranked Attributes.



Acc_no	Account	PerId	Person	service	flag	src_bytes	dst_bytes	logged_in	count	serror_rat	host_srv_	st_same_s	st_diff_sr	srv_diff_	t_srv_ser	class
Acc1	Account	P1	Person	7	0	491	0	0	2	0	25	0.17	0.03	0	0	0
Acc2	Account	P2	Person	9	0	146	0	0	13	0	1	0	0.6	0	0	0
Acc3	Account	P3	Person	2	1	0	0	0	123	1	26	0.1	0.05	0	1	1
Acc4	Account	P4	Person	4	0	232	8153	1	5	0.2	255	1	0	0.04	0.01	0
Acc5	Account	P5	Person	4	0	199	420	1	30	0	255	1	0	0	0	0
Acc6	Account	P6	Person	2	2	0	0	0	121	0	19	0.07	0.07	0	0	1
Acc7	Account	P7	Person	2	1	0	0	0	166	1	9	0.04	0.05	0	1	1
Acc8	Account	P8	Person	2	1	0	0	0	117	1	15	0.06	0.07	0	1	1
Acc9	Account	P9	Person	13	1	0	0	0	270	1	23	0.09	0.05	0	1	1
Acc10	Account	P10	Person	2	1	0	0	0	133	1	13	0.05	0.06	0	1	1
Acc11	Account	P11	Person	2	2	0	0	0	205	0	12	0.05	0.07	0	0	1
Acc12	Account	P12	Person	2	1	0	0	0	199	1	13	0.05	0.07	0	1	1
Acc13	Account	P13	Person	4	0	287	2251	1	3	0	219	1	0	0.03	0	0
Acc14	Account	P14	Person	7	0	334	0	1	2	0	20	1	0	0.2	0	1
Acc15	Account	P15	Person	15	1	0	0	0	233	1	1	0	0.07	0	1	1
Acc16	Account	P16	Person	14	1	0	0	0	96	1	2	0.01	0.06	0	1	1
Acc17	Account	P17	Person	4	0	300	13788	1	8	0	255	1	0	0.02	0	0
Acc18	Account	P18	Person	10	0	18	0	0	1	0	16	1	0	1	0	1
Acc19	Account	P19	Person	4	0	233	616	1	3	0	255	1	0	0.03	0	0
Acc20	Account	P20	Person	4	0	343	1178	1	9	0	255	1	0	0.04	0	0
Acc21	Account	P21	Person	16	1	0	0	0	223	1	23	0.09	0.05	0	1	1
Acc22	Account	P22	Person	2	1	0	0	0	280	1	17	0.07	0.06	0	1	1
Acc23	Account	P23	Person	4	0	253	11905	1	8	0	255	1	0	0.02	0	0
Acc24	Account	P24	Person	9	0	147	105	0	1	0	1	0	0.85	0	0	0
Acc25	Account	P25	Person	16	1	0	0	0	248	1	2	0.01	0.06	0	1	1
Acc26	Account	P26	Person	0	0	437	14421	1	1	0	25	0.1	0.05	0	0	0

Figure (4.2) Dataset after preparation.

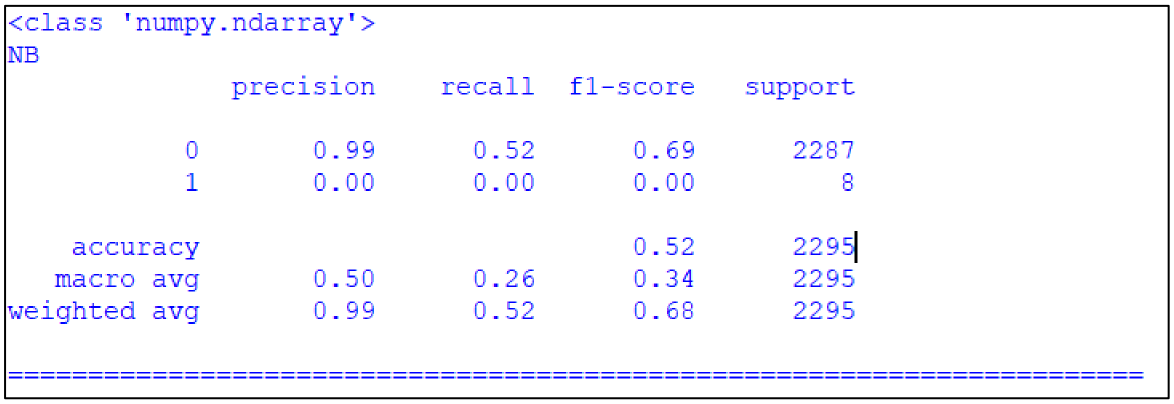
4.6 Threshold Calculated

To calculate the threshold values, network intrusion detection dataset is fed to a set of machine learning classification algorithms including Naive Bayes (NB), k-nearest neighbors (KNN), Decision Tree (DT), Stochastic gradient descent (SGD), and Logistic regression (LR). Accuracy is used for performance evaluation as shown in Figures (4.3), (4.4), (4.5), (4.6). In the experiments, Decision Tree has shown the best performance of 99% than others algorithm as shown in Figure (4.7).

Explain the outcomes of the decision tree algorithm (J48) for Network Intrusion Dataset in Table (4.6). The DT model programed in python programing language as shown in Figure (4.9). Each of those features has a specific threshold that is extracted from the DT rule. A threshold value determines the previous probabilities could be interpret as class labels to be classified as binary scoring. The threshold is the range of criteria values which determine the positive conditions that provide the maximal rate of accuracy. A threshold value is decided by J48 as explained in algorithm (1) in chapter3.

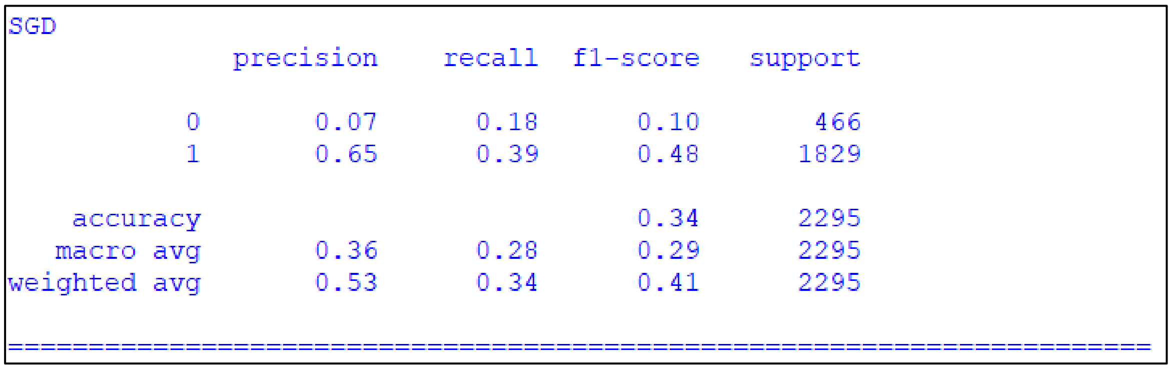
Table (4.6) The most important features and their Threshold (Cut-off points) of dataset.

No.	Features name	Cut-off point
1	service	8
2	falg	5.5
3	src_bytes	28.5
4	dst_bytes	1
5	logged_in	0
6	Count	2
7	serror_rate	0.73
8	dst_host_srv_count	89
9	dst_host_same_srv_rate	0.53
10	dst_host_diff_srv_rate	0.89
11	dst_host_srv_diff_host_rate	0.25
12	dst_host_srv_serror_rate	0.79



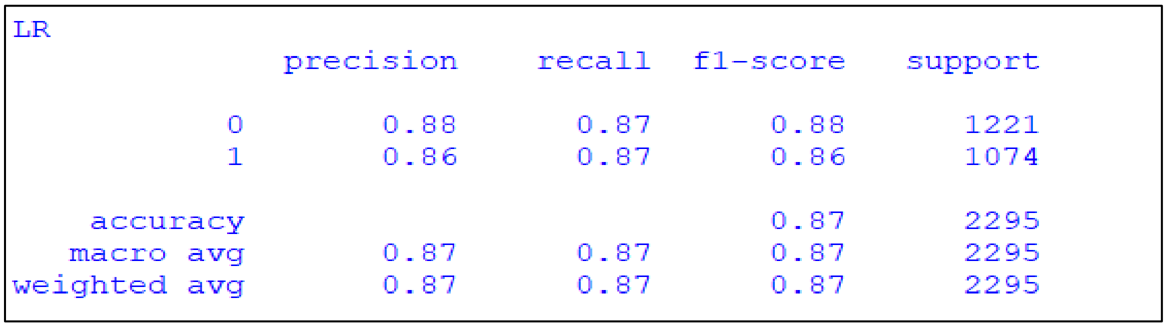
```
<class 'numpy.ndarray'>
NB
              precision    recall  f1-score   support

           0       0.99      0.52      0.69      2287
           1       0.00      0.00      0.00         8

    accuracy                           0.52      2295
   macro avg       0.50      0.26      0.34      2295
weighted avg       0.99      0.52      0.68      2295

========================================================================
```

Figure (4.3) Naive Bayes results for calculating threshold.



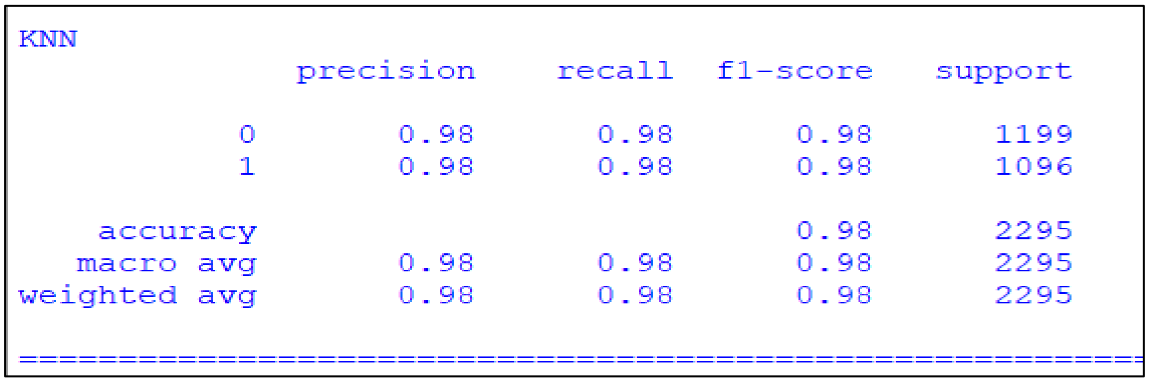
```
SGD
              precision    recall  f1-score   support

           0       0.07      0.18      0.10       466
           1       0.65      0.39      0.48      1829

    accuracy                           0.34      2295
   macro avg       0.36      0.28      0.29      2295
weighted avg       0.53      0.34      0.41      2295

========================================================================
```

Figure (4.4) Stochastic gradient descent results for calculating threshold.



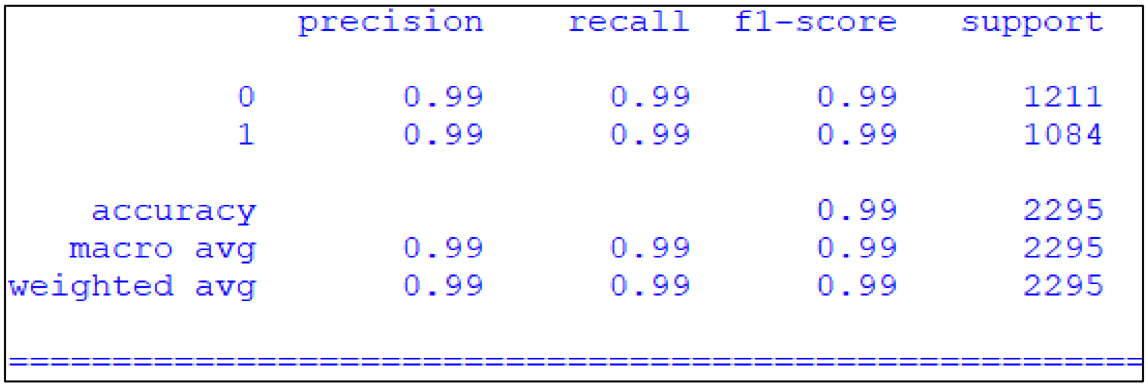
```
LR
              precision    recall  f1-score   support

           0       0.88      0.87      0.88      1221
           1       0.86      0.87      0.86      1074

    accuracy                           0.87      2295
   macro avg       0.87      0.87      0.87      2295
weighted avg       0.87      0.87      0.87      2295
```

Figure (4.6) Logistic regression results for calculating threshold.



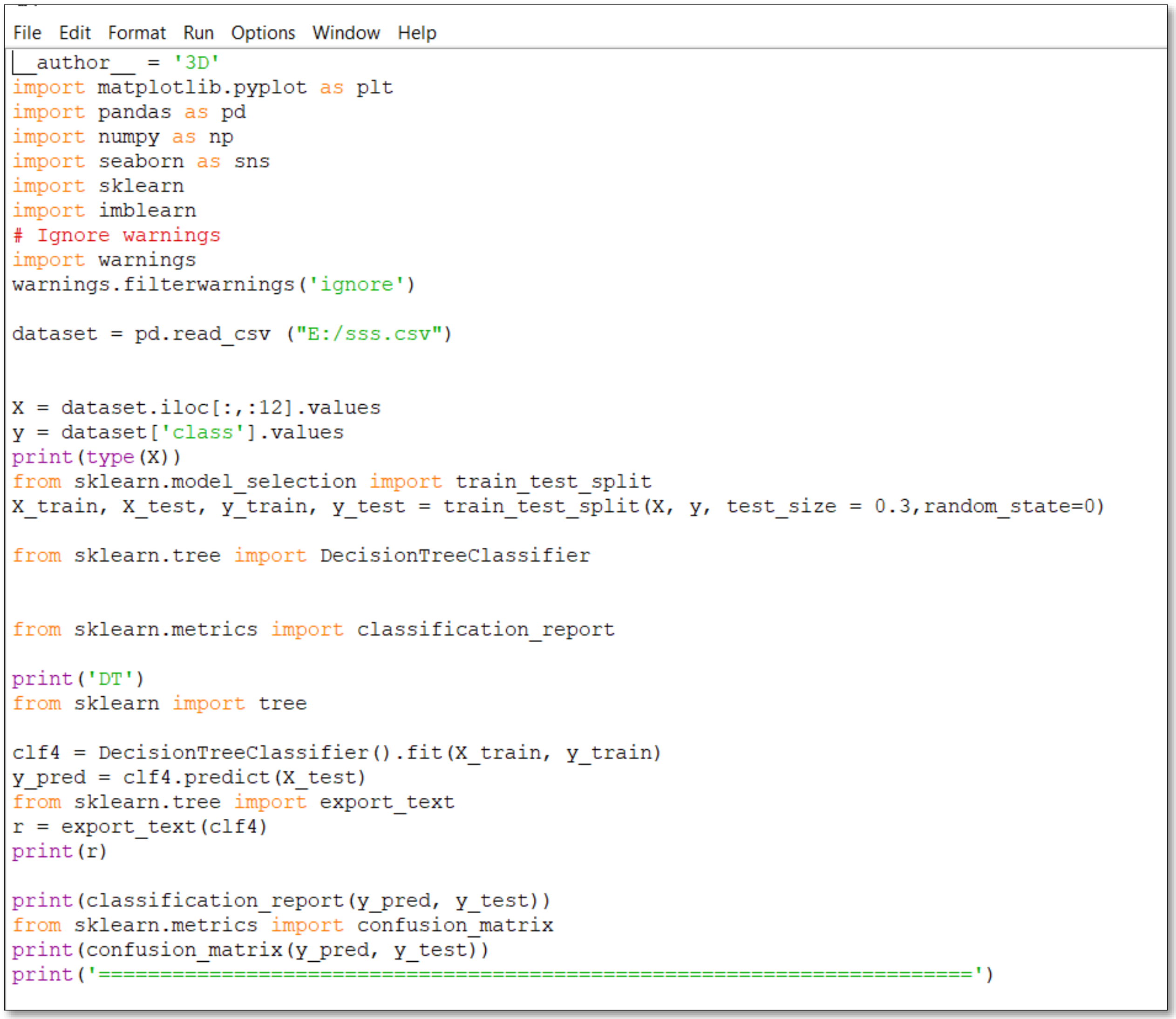
```
KNN
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1199
           1       0.98      0.98      0.98      1096

    accuracy                           0.98      2295
   macro avg       0.98      0.98      0.98      2295
weighted avg       0.98      0.98      0.98      2295

=========================================================
```

Figure (4.7) K-nearest neighbors results for calculating threshold.



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1211
           1       0.99      0.99      0.99      1084

    accuracy                           0.99      2295
   macro avg       0.99      0.99      0.99      2295
weighted avg       0.99      0.99      0.99      2295

=========================================================
```

Figure (4.8) Decision tree results for calculating threshold.

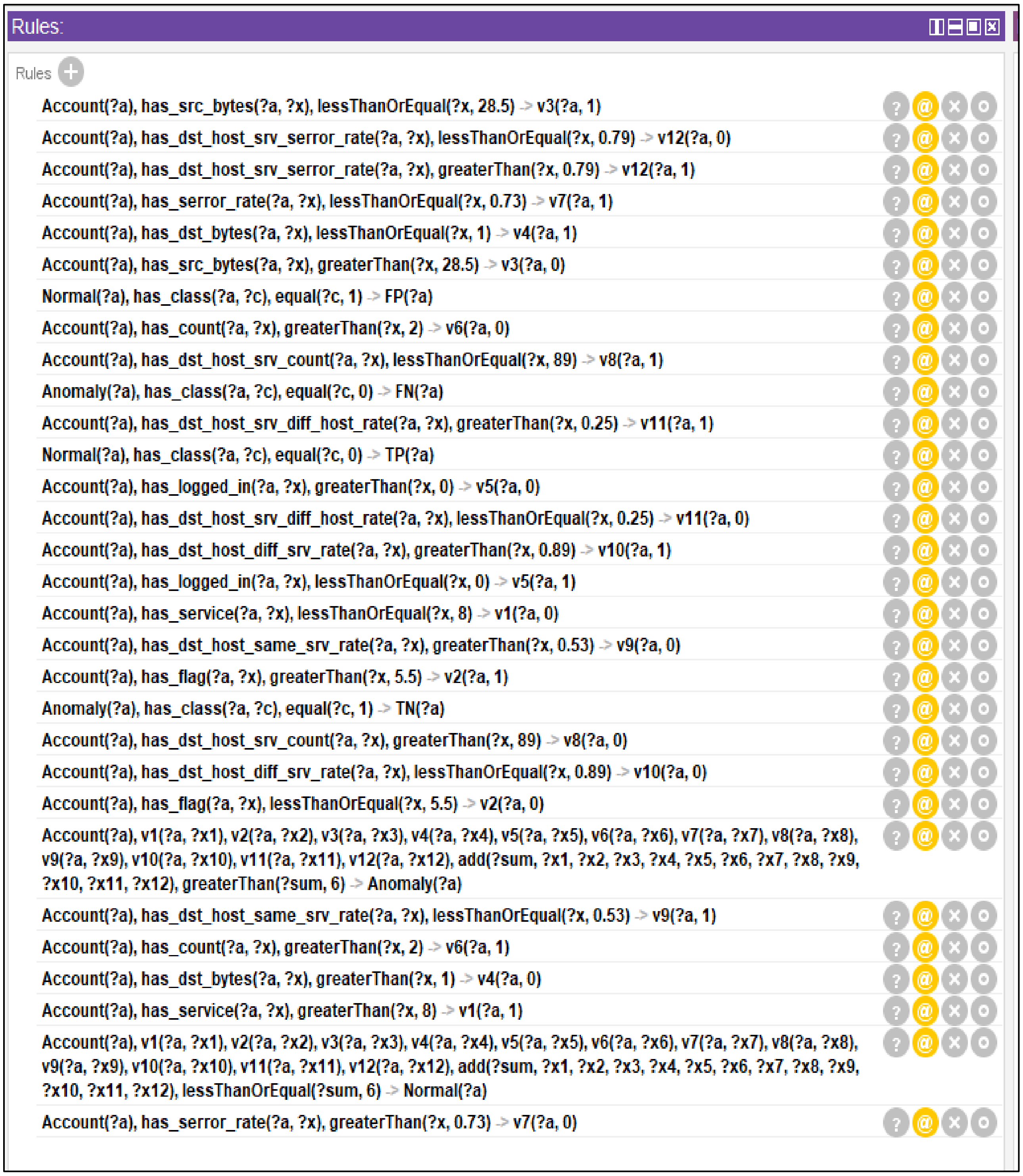


```
__author__ = '3D'
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
import sklearn
import imblearn
# Ignore warnings
import warnings
warnings.filterwarnings('ignore')

dataset = pd.read_csv ("E:/sss.csv")


X = dataset.iloc[:,:12].values
y = dataset['class'].values
print(type(X))
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,random_state=0)

from sklearn.tree import DecisionTreeClassifier


from sklearn.metrics import classification_report

print('DT')
from sklearn import tree

clf4 = DecisionTreeClassifier().fit(X_train, y_train)
y_pred = clf4.predict(X_test)
from sklearn.tree import export_text
r = export_text(clf4)
print(r)

print(classification_report(y_pred, y_test))
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_pred, y_test))
print('========================================================================')
```

Figure (4.9) the decision tree model for calculating threshold.

4.7 SWRL Rules Creation

SWRL rules are used to derive new information from the present knowledge. In terms of ontological concepts, all rules are expressed such as classes, properties, and individuals. A threshold is a mandatory factor in creating semantic web rules used by the reasoner to differentiate anomalies. Figure (4.10) showing OWL Plugin Active Ontology Tab.

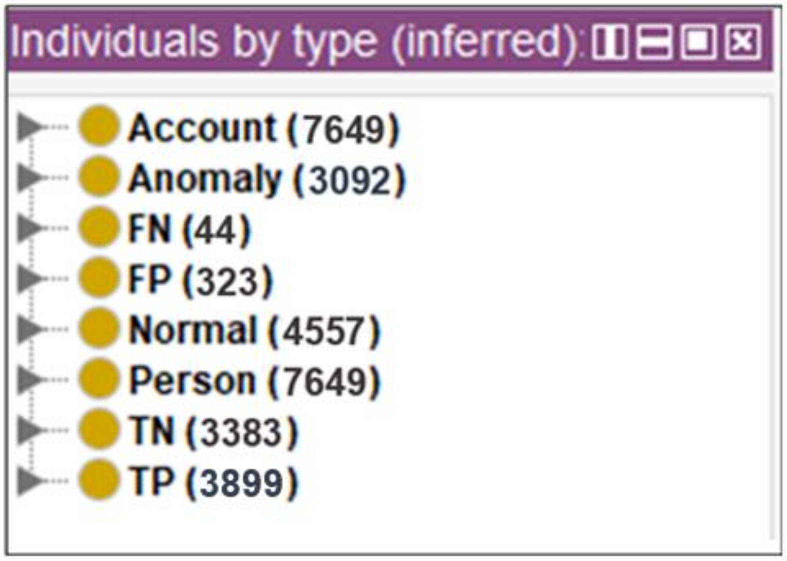


```
Account(?a), has_src_bytes(?a, ?x), lessThanOrEqual(?x, 28.5) -> v3(?a, 1)
Account(?a), has_dst_host_srv_serror_rate(?a, ?x), lessThanOrEqual(?x, 0.79) -> v12(?a, 0)
Account(?a), has_dst_host_srv_serror_rate(?a, ?x), greaterThan(?x, 0.79) -> v12(?a, 1)
Account(?a), has_serror_rate(?a, ?x), lessThanOrEqual(?x, 0.73) -> v7(?a, 1)
Account(?a), has_dst_bytes(?a, ?x), lessThanOrEqual(?x, 1) -> v4(?a, 1)
Account(?a), has_src_bytes(?a, ?x), greaterThan(?x, 28.5) -> v3(?a, 0)
Normal(?a), has_class(?a, ?c), equal(?c, 1) -> FP(?a)
Account(?a), has_count(?a, ?x), greaterThan(?x, 2) -> v6(?a, 0)
Account(?a), has_dst_host_srv_count(?a, ?x), lessThanOrEqual(?x, 89) -> v8(?a, 1)
Anomaly(?a), has_class(?a, ?c), equal(?c, 0) -> FN(?a)
Account(?a), has_dst_host_srv_diff_host_rate(?a, ?x), greaterThan(?x, 0.25) -> v11(?a, 1)
Normal(?a), has_class(?a, ?c), equal(?c, 0) -> TP(?a)
Account(?a), has_logged_in(?a, ?x), greaterThan(?x, 0) -> v5(?a, 0)
Account(?a), has_dst_host_srv_diff_host_rate(?a, ?x), lessThanOrEqual(?x, 0.25) -> v11(?a, 0)
Account(?a), has_dst_host_diff_srv_rate(?a, ?x), greaterThan(?x, 0.89) -> v10(?a, 1)
Account(?a), has_logged_in(?a, ?x), lessThanOrEqual(?x, 0) -> v5(?a, 1)
Account(?a), has_service(?a, ?x), lessThanOrEqual(?x, 8) -> v1(?a, 0)
Account(?a), has_dst_host_same_srv_rate(?a, ?x), greaterThan(?x, 0.53) -> v9(?a, 0)
Account(?a), has_flag(?a, ?x), greaterThan(?x, 5.5) -> v2(?a, 1)
Anomaly(?a), has_class(?a, ?c), equal(?c, 1) -> TN(?a)
Account(?a), has_dst_host_srv_count(?a, ?x), greaterThan(?x, 89) -> v8(?a, 0)
Account(?a), has_dst_host_diff_srv_rate(?a, ?x), lessThanOrEqual(?x, 0.89) -> v10(?a, 0)
Account(?a), has_flag(?a, ?x), lessThanOrEqual(?x, 5.5) -> v2(?a, 0)
Account(?a), v1(?a, ?x1), v2(?a, ?x2), v3(?a, ?x3), v4(?a, ?x4), v5(?a, ?x5), v6(?a, ?x6), v7(?a, ?x7), v8(?a, ?x8),
v9(?a, ?x9), v10(?a, ?x10), v11(?a, ?x11), v12(?a, ?x12), add(?sum, ?x1, ?x2, ?x3, ?x4, ?x5, ?x6, ?x7, ?x8, ?x9,
?x10, ?x11, ?x12), greaterThan(?sum, 6) -> Anomaly(?a)
Account(?a), has_dst_host_same_srv_rate(?a, ?x), lessThanOrEqual(?x, 0.53) -> v9(?a, 1)
Account(?a), has_count(?a, ?x), greaterThan(?x, 2) -> v6(?a, 1)
Account(?a), has_dst_bytes(?a, ?x), greaterThan(?x, 1) -> v4(?a, 0)
Account(?a), has_service(?a, ?x), greaterThan(?x, 8) -> v1(?a, 1)
Account(?a), v1(?a, ?x1), v2(?a, ?x2), v3(?a, ?x3), v4(?a, ?x4), v5(?a, ?x5), v6(?a, ?x6), v7(?a, ?x7), v8(?a, ?x8),
v9(?a, ?x9), v10(?a, ?x10), v11(?a, ?x11), v12(?a, ?x12), add(?sum, ?x1, ?x2, ?x3, ?x4, ?x5, ?x6, ?x7, ?x8, ?x9,
?x10, ?x11, ?x12), lessThanOrEqual(?sum, 6) -> Normal(?a)
Account(?a), has_serror_rate(?a, ?x), greaterThan(?x, 0.73) -> v7(?a, 0)
```

Figure (4.10) OWL Plugin Active Ontology Tab showing all SWRL rules.

4.8 The Reasoner of Ontology SFW

The ontology reasoner focuses on semantic web rules (logic laws). The semantic criteria are developed by means of the use of threshold values that can recognize and categorize anomalies. While data preparation and feature selection are a critical step in the model evaluation, it is equally important to measure the performance of each feature. Various metrics are used to evaluate the performance. As for the proposed model's overall prediction capabilities, it needs to be improved before applying it onto unseen. The appropriate model assessment uses differing metrics like accuracy, precision, recall, and f1-score as shown in Figure (4.11). Ontology IRI (http://www.semanticweb.org/Baraa/ontologies/2021/5/SFW).



Account (7649)
Anomaly (3092)
FN (44)
FP (323)
Normal (4557)
Person (7649)
TN (3383)
TP (3899)

Figure (4.11) Reasoner results of SFW approach.

4.9 Model Evaluation

The evaluation outcomes prove that the ontology model gives better results than other existing ML-based models. The efficiency of the OWL ontology model is determined using measures like precision, recall and F-measure. These measures used to evaluate the performance of OWL ontology models in comparison to human judgments. The accuracy of the ontology model in classifying anomalies reaches a percentage of (95%). In addition to that, the ontology classifier is considered to be a more comprehensive model for providing direct and human-interpretable decision rules. The evaluation metrics clarify that the ontology meter models accuracy is very close to or even achieves better performance than machine learning techniques of other related works.

A confusion matrix is an A × A matrix where A is the class number. There are two classes: anomaly and normal. The columns of the matrix represent real classes, and the rows represent predicted classes. The confusion matrix gives the numbers of correctly and incorrectly predicted results by the model. Tables (4.7) and (4.8) explain the confusion matrix of the proposed model. And Table (4.9), Figure (4.12) explain the Evaluation metrics of semantic Firewall (SFW) ontology approach.

Predicated / Actually	Normal	Attack
Normal	**TP**	**FN**
Attack	**FP**	**TN**

Table (4.7) Confusion matrix.

Predicated / Actually	Normal	Attack
Normal	**3899**	**44**
Attack	**323**	**3383**

Table (4.8) Confusion matrix of SFW ontology approach.

Table (4.9) Evaluation metrics of SFW ontology approach.

Technique	Accuracy	Precision	Recall	F-M	Error rate
Ontology SFW	0.952	0.923	0.988	0.944	0.05

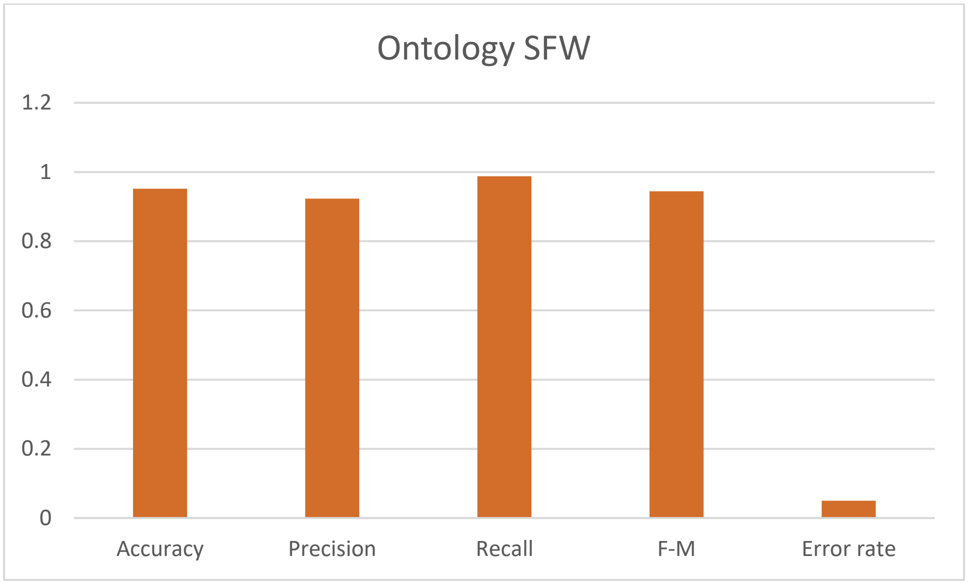


Figure (4.12) Evaluation metrics of SFW.

4.10 Discussion

If we discuss the results from interpretable and clarity (How & Why), the ontology outcomes are considered the best because the decision making is understandable and easily described. Most machine learning algorithms are "semi black boxes" to make excellent predictions, but that the logic behind those predictions is somewhat ambiguous. In contrast, our ontology classifier is an interpretable model, as shown in Figure (4.13).

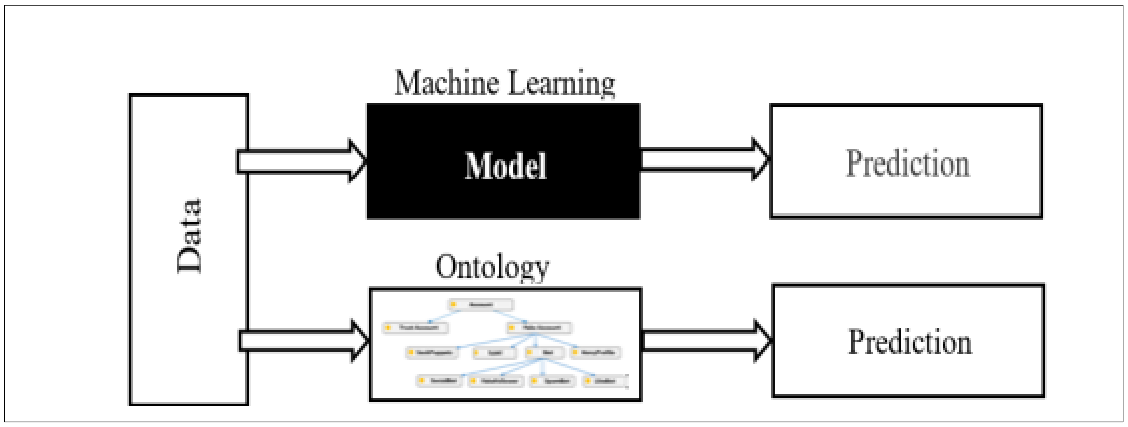


Figure (4.13). Machine learning classifier is semi-black box while Ontology classifier is an interpretable model.

4.11 summary

This chapter covered the experimental environment, practical explanation of data pre-processing and feature selection, as well as the results of algorithms in the proposed model of the datasets (Network Intrusion Detection). Such an ontology is a common taxonomy. The future network will require the ontological contents of all current objects. This will help researchers and developers organize their issues in form of ontologies.

Chapter Five

Conclusions and Future Work

5.1 Conclusions

The development of ontology-based applications is becoming easier as semantic web approaches, languages, and tools evolve. The work presented in this thesis shows that SWRL rules and domain ontology can be used to construct a compelling prototype of a semantic firewall. The foundation of the proposed method is the creation of a firewall domain ontology to which logical inference rules are applied. Protégé editor is the most straightforward framework for ontology developers. Moreover, it is rich with several plugins for visualization, inferring, and querying, making it flexible and powerful. The following show some conclusion :

- To predict anomalies, SWRL rules are derived from strong connections between the ontology concepts.
- The functions of the proposed approach are divided into four tasks, namely, data preparation, classification and relationship, ontology building, and finally the application of SWRL rules to infer whether or not the record is an anomaly.
- A threshold is a mandatory factor for creating SWRL rules. The reasoners use these rules to differentiate an anomaly record from normal records. It has been found that the used machine learning method with the decision tree algorithm is perfect for determining the threshold for SWRL rules.
- The reasoner makes use of data and rules to draw the inference and provide the final decision, showing whether the data is an anomaly or normal.
- Experiments are carried out to test the approach performance in recognizing 7650 records. The system evaluation depends on SWRL rules and standard evaluation metrics, and results show that the proposed approach can correctly identify 4076 normal and 3573 anomalies, amounting to an accuracy of 95%.

- This work's primary contribution is the use of SWRL rules and ontology to identify attacks. Additionally, an ontology classifier is a readable model with simple and human-interpretable decision rules.

5.2 Future work

A number of future works can be carried out as summarized in the following points:

1. Building a firewall which can detect more types of attacks in different OSI layers.
2. Modifying and tuning a classification model threshold to balance better outcomes of a related model using different approaches.
3. Optimizing the features to increase the accuracy rate with other datasets.

References

[1] I. Kashefi, M. Kassiri, and A. Shahidinijad, “A survey of on security issues in firewall: a new approach for classifying fire wall vulnerabilieties,” Int. J. Eng. Researh Appl., vol. 3, no. 2, pp. 585–591, 2013.

[2] C. A. Fowler and R. J. Hammell II, “Building baseline preprocessed common data sets for multiple follow-on data mining algorithms,” in Proceedings of the Conference on Information Systems Applied Research ISSN, 2012.

[3] C. Apté and S. Weiss, “Data mining with decision trees and decision rules,” Futur. Gener. Comput. Syst., vol. 13, no. 2–3, pp. 197–210, 1997.

[4] R. Ashri, T. Payne, D. Marvin, M. Surridge, and S. Taylor, “Towards a semantic web security infrastructure,” 2004.

[5] A.-M. Ghiran, G. C. Silaghi, and N. Tomai, “Ontology-based tools for automating integration and validation of firewall rules,” in International Conference on Business Information Systems, 2009, pp. 37–48.

[6] T. Chomsiri, “Tree Rule Firewall,” no. November, 2016.

[7] M. H. Jabardi and A. S. Hadi, “Using Machine Learning to Inductively Learn Semantic Rules,” J. Phys. Conf. Ser., vol. 1804, no. 1, 2021.

[8] Q. Ismail, O. Saleh, M. Hashayka, A. Awad, A. Hawash, and O. Othman, “Improve the firewall accuracy by using dynamic ontology,” PervasiveHealth Pervasive Comput. Technol. Healthc., 2020.

[9] Hajar, H. Qasem, S.D.Khamitkar “Discovering-Anomalous-Rules-In-Firewall-Logs-Using-Data-Mining-And-Machine-Learning-Classifiers, pp. 2277-8616 2020.

[10] R. F. Cordova, A. L. Marcovich, and C. A. Santivanez, “An Efficient Method for Ontology-Based Multi-Vendor Firewall Misconfiguration

Detection: A Real-Case Study," 2018 IEEE ANDESCON, Conf. Proc., pp. 1–3, 2018.

[11] J. Vincent, C. Porquet, M. Borsali, and H. Leboulanger, "Privacy Protection for Smartphones: An Ontology-Based Firewall," Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication Lecture Notes in Computer Science, pp. 371–380, 2011.

[12] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin, Firewalls and Internet security: repelling the wily hacker. Addison-Wesley Professional, 2003.

[13] T. H. Hadi, "How Firewall Keeps The Signal Save In Local LAN ?," no. October, 2019.

[14] S. K. V. Sharma, "A Security Solution for Wireless Local Area Network (WLAN) Using Firewall and VPN‖," Innov. Syst. Des. Eng., vol. 7, no. 7, 2016.

[15] M. N. Bin Ali, M. E. Hossain, and M. M. Parvez, "Design and implementation of a secure campus network," Int. J. Emerg. Technol. Adv. Eng., vol. 5, no. 7, pp. 370–374, 2015.

[16] A. Chopra, "Security issues of firewall," Int. J. P2P Netw. Trends Technol., vol. 22, no. 1, pp. 4–9, 2016.

[17] K. Das and R. N. Behera, "A survey on machine learning: concept, algorithms and applications," Int. J. Innov. Res. Comput. Commun. Eng., vol. 5, no. 2, pp. 1301–1309, 2017.

[18] N. Dutta, S. Umashankar, V. K. A. Shankar, S. Padmanaban, Z. Leonowicz, and P. Wheeler, "Centrifugal pump cavitation detection using machine learning algorithm technique," in 2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE

Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2018, pp. 1–6.

[19] D. Dhall, R. Kaur, and M. Juneja, “Machine learning: a review of the algorithms and its applications,” Proc. ICRIC 2019, pp. 47–63, 2020.

[20] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” Synth. Lect. Artif. Intell. Mach. Learn., vol. 3, no. 1, pp. 1–130, 2009.

[21] S. Sahu and B. M. Mehtre, “Network intrusion detection system using J48 Decision Tree,” in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 2023–2026.

[22] H. Sharma and S. Kumar, “A survey on decision tree algorithms of classification in data mining,” Int. J. Sci. Res., vol. 5, no. 4, pp. 2094–2097, 2016.

[23] P.-N. Tan, M. Steinbach, and V. Kumar, “Classification: basic concepts, decision trees, and model evaluation,” Introd. to data Min., vol. 1, pp. 145–205, 2006.

[24] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised leaning,” Int. J. Comput. Sci., vol. 1, no. 2, pp. 111–117, 2006.

[25] M. A. Hall, “Correlation-based feature selection of discrete and numeric class machine learning,” 2000.

[26] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” Sci. Am., vol. 284, no. 5, pp. 34–43, 2001.

[27] I. Ullah, The role of semantic web technologies for IoT data in underpinning environmental science. Lancaster University (United Kingdom), 2018.

[28] N. Guarino, Formal ontology in information systems: Proceedings of the first international conference (FOIS’98), June 6-8, Trento, Italy, vol. 46. IOS press, 1998.

[29] S. Bhosale and I. Mumbai, Maharashtra, “Network Intrusion Detection,” 2017. https://www.kaggle.com/sampadab17/network-intrusion-detection.

[30] O. Signore, “Representing knowledge in the semantic web,” in Open Culture Conference (organised by the Italian office of W3C), 2005, pp. 27–29.

[31] S. Bratt, “Semantic web and other W3C technologies to watch,” Talks W3C, January, 2007.

[32] A. Gerber, A. Van der Merwe, and A. Barnard, “A functional semantic web architecture,” in European Semantic Web Conference, 2008, pp. 273–287.

[33] A. J. Gerber, A. Barnard, and A. J. Van der Merwe, “Towards a semantic web layered architecture,” 2007.

[34] N. F. Noy and D. L. McGuinness, “Ontology development 101: A guide to creating your first ontology.” Stanford knowledge systems laboratory technical report KSL-01-05 and …, 2001.

[35] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A semantic web rule language combining OWL and RuleML,” W3C Memb. Submiss., vol. 21, no. 79, pp. 1–31, 2004.

[36] R. Goebel, S. Zilles, C. Ringlstetter, A. Dengel, and G. A. Grimnes, “What is the Role of the Semantic Layer Cake for Guiding the Use of Knowledge Representation and Machine Learning in the Development of the Semantic Web?,” in AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering, 2008, pp. 45–50.

[37] H. Waguih, “A proposed trust model for the semantic web,” 2006.

[38] Y. Zhang, H. Chen, Z. Wu, and X. Zheng, “A reputation-chain trust model for the semantic Web,” in 20th International Conference on Advanced

Information Networking and Applications-Volume 1 (AINA'06), 2006, vol. 2, pp. 5-pp.

[39] H. Hendi, "Ontologies and semantic web for an evolutive development of logistic applications," 2017.

[40] J. D. Allen et al., "The unicode standard," Mt. view, CA, 2012.

[41] D. Beckett and B. McBride, "RDF/XML syntax specification (revised)," W3C Recomm., vol. 10, no. 2.3, 2004.

[42] Q. Tong, "Mapping object-oriented database models into RDF (S)," IEEE Access, vol. 6, pp. 47125–47130, 2018.

[43] C. McGhee. (2018, 10/3/2020). What is Semantic Web and How Does it Work? Available: https://www.mobidea.com/academy/what-is-semantic-web/.

[44] C. Reyes-Pena and M. Tovar-Vidal, "Ontology: Components and Evaluation, a Review," Res. Comput. Sci., vol. 148, pp. 257–265, 2019.

[45] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," ACM Comput. Surv., vol. 51, no. 5, pp. 1–42, 2018.

[46] M. Keet, An introduction to ontology engineering, vol. 1. Maria Keet Cape Town, 2018.

[47] R. Power and A. Third, "Expressing OWL axioms by English sentences: dubious in theory, feasible in practice," 2010.

[48] M. Donnelly and G. Guizzardi, Formal Ontology in Information Systems: Proceedings of the Seventh International Conference (FOIS 2012), vol. 239. IOS Press, 2012.

[49] P. F. Patel-Schneider, "OWL web ontology language semantics and abstract syntax, W3C Recommendation," http//www. w3. org/TR/2004/REC-owl-semantics-20040210/, 2004.

[50] C. Welty, D. L. McGuinness, and M. K. Smith, “Owl web ontology language guide,” W3C Recomm. W3C (February 2004) http//www. w3. org/TR/2004/REC-owl-guide-20040210, p. 48, 2004.

[51] M. Horridge, “Owl syntaxes,” Ontogenesis, 2010.

[52] C. Bock et al., “Structural Specification and Functional-Style Syntax,” 2008.

[53] M. Horridge and P. F. Patel-Schneider, “OWL 2 web ontology language manchester syntax,” W3C Work. Gr. Note, 2009.

[54] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, “SWRL: A semantic web rule language combining OWL and RuleML,” W3C Memb. Submiss., vol. 21, no. 79, pp. 1–31, 2004.

[55] S. Decker et al., “The semantic web: The roles of XML and RDF,” IEEE Internet Comput., vol. 4, no. 5, pp. 63–73, 2000.

[56] T. Segaran, C. Evans, and J. Taylor, Programming the Semantic Web: Build Flexible Applications with Graph Data. “ O’Reilly Media, Inc.,” 2009.

[57] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, and A. Steigmiller, “The OWL reasoner evaluation (ORE) 2015 competition report,” J. Autom. Reason., vol. 59, no. 4, pp. 455–482, 2017.

[58] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, “Pellet: A practical owl-dl reasoner,” J. Web Semant., vol. 5, no. 2, pp. 51–53, 2007.

[59] S. Abburu, “A survey on ontology reasoners and comparison,” Int. J. Comput. Appl., vol. 57, no. 17, 2012.

[60] B. Glimm, I. Horrocks, B. Motik, R. Shearer, and G. Stoilos, “A novel approach to ontology classification,” J. Web Semant., vol. 14, pp. 84–101, 2012.

[61] L. Obrst, “Ontologies for semantically interoperable systems,” in Proceedings of the twelfth international conference on Information and knowledge management, 2003, pp. 366–369.

[62] J.-M. López-Cobo, S. Losada, L. Cicurel, J. L. Bas, S. Bellido, and R. Benjamins, “Ontology management in e-banking applications,” in Ontology management, Springer, 2008, pp. 229–244.

[63] C. A. Kumar, M. P. Sooraj, and S. Ramakrishnan, “A comparative performance evaluation of supervised feature selection algorithms on microarray datasets,” Procedia Comput. Sci., vol. 115, pp. 209–217, 2017.

[64] R. Kirkby and E. Frank, “WEKA Explorer User Guide for Version 3-4,” Univ. Weikato, pp. 3–4, 2002.

[65] S. Dinakaran and P. R. J. Thangaiah, “Role of attribute selection in classification algorithms,” Int. J. Sci. Eng. Res., vol. 4, no. 6, pp. 67–71, 2013.

[66] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, “Data preprocessing for supervised leaning,” Int. J. Comput. Sci., vol. 1, no. 2, pp. 111–117, 2006.

[67] M. d’Aquin and N. F. Noy, “Where to publish and find ontologies? A survey of ontology libraries,” J. Web Semant., vol. 11, pp. 96–111, 2012.

[68] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, “Confusion matrix-based feature selection.,” MAICS, vol. 710, pp. 120–127, 2011.

[69] M. M. Taye, “Understanding semantic web and ontologies: Theory and applications,” arXiv Prepr. arXiv1006.4567, 2010.

[70] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” Pattern Recognit., vol. 91, pp. 216–231, 2019.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل كلية تكنولوجيا المعلومات

قسم شبكات المعلومات

قواعد جدار الحماية الدلالية المستندة إلى علم الوجود لحماية الشبكة

مقدمة الى مجلس كلية تكنولوجيا المعلومات جامعة بابل وهي جزء من متطلبات نيل درجة الماجستير في تكنولوجيا المعلومات / شبكات المعلومات

من قبل الطالبة

براء هاشم كريم فجر

أشراف

أ.د . وسام سمير عبدعلي بهيه

1443 هـ

2021 م

الخلاصة

يتزايد حجم وتنوع هجمات الشبكة أثناء استخدام نفس تقنيات الهجوم الدفاعي الأساسية. جدار الحماية هو أحد مكونات مفتاح أمان الشبكة الذي يقوم بتصفية حزم الشبكة الواردة والصادرة وفقًا لقواعد الأمان المحددة مسبقًا. على الرغم من أن جدران الحماية هي وسيلة دفاع فعالة ضد بعض الهجمات، إلا أنها تحتوي على عيوب أمنية معينة يمكن الاستفادة منها في ظروف أخرى. يمكن لجدار الحماية القديم تجاوز اختراق الشبكة غير الطبيعي. بالنسبة لاقتحام الشبكة غير الطبيعي في جدار الحماية الجديد، يُزعم أن جدار الحماية الدلالي المستند إلى علم الوجود وخوارزميات التعلم الآلي يمكن أن يعزز جدار الحماية ويحمي الشبكة بشكل فعال. اقترحت هذه الرسالة نموذجًا قائمًا على علم الوجود كجدار حماية دلالي كمحاولة لاستكشاف فعاليته. تعتمد الطريقة المقترحة في هذه الرسالة على الوصف المنطقي العقلاني (DLR) وواجهات برمجة تطبيقات الأنطولوجيا مع لغات الويب الدلالية (OWL وSWRL) المدعومة بالتعلم الآلي. يتخذ جدار الحماية الدلالي المقترح قراراته بشأن الكشف عن الحالات الشاذة بناءً على مجموعة من قواعد الحماية للنموذج القائم على علم الوجود. تُستخدم خوارزمية شجرة القرار لحساب العتبة التي تُعد العمود الفقري للمُبررات المستندة إلى القواعد لـ SWRL والتي تُستخدم في الكشف عن الحالات الشاذة. استنادًا إلى قواعد SWRL، يمكن للمسبب إعطاء القرار. نتيجة لذلك، يحقق النهج المقترح دقة كشف تبلغ 95٪. أخيرًا، تم التوصل إلى استنتاج مفاده أن مصنف الأنطولوجيا يقدم نموذجًا شاملاً لجدار الحماية الدلالي الذي يوفر قواعد قرار صريحة وقابلة للتفسير البشري، مقابل نماذج التعلم الآلي البديلة.