إقــــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

نظام كشف التسلل باستخدام خوارزمية تقارب الانتشار المعدلة وخوارزميات التصنيف

Intrusion Detection System using Improved Affinity Propagation and Classification

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name:

Signature:

Date:

اسم الطالب: أحمد معين محمد سرداح

التوقيع:

التاريخ: 14 / 11 / 2015

Islamic University of Gaza
Research & Graduate Affairs
Faculty of Engineering
Computer Engineering Department

الجامعة الاسلامية – غزة
شئون البحث العلمي و الدراسات العليا
كلية الهندسة
قسم هندسة الحاسوب

Intrusion Detection System using Improved Affinity Propagation and Classification

By

Ahmed M. Serdah

120110627

Supervisor:

Dr. Wesam M. Ashour

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Computer Engineering

(1437هـ – 2015م)

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

مكتب نائب الرئيس للبحث العلمي والدراسات العليا هاتف داخلي 1150

الرقم س غ/35/ Ref ....

التاريخ 2015/11/02 Date ....

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة شئون البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ أحمد معين محمد سرداح لنيل درجة الماجستير في كلية *الهندسة* قسم هندسة الحاسوب وموضوعها:

نظام كشف التسلل باستخدام خورازمية تقارب الانتشار المعدلة وخورازميات التصنيف

Intrusion Detection System using Improved Affinity Propagation and Classification

وبعد المناقشة التي تمت اليوم الاثنين 20 محرم 1437هـ، الموافق 2015/11/02م الساعة الحادية عشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. وسام محمود عاشور	مشرفاً و رئيساً	....
د. محمد أحمد الحنجوري	مناقشاً داخلياً	Alhenjoun ....
د. إيهاب صلاح زقوت	مناقشاً خارجياً	....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية *الهندسة* / قسم وموضوعها: هندسة الحاسوب.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق ،،،

نائب الرئيس لشئون البحث العلمي والدراسات العليا

أ.د. عبد الرؤوف علي المناعمة

Dedication

I would like to dedicate this study to:

All those, who taught me a letter;
My beloved mother whose love, care, support and inspired me to reach this far;
My father who has always loved and supported me, throughout all my life;
My sincere wife who give me a continuous support and understanding;
My little daughters wishing them a great future;
My brothers and sisters for their encouragement and support;
My wonderful supervisor he was my inspiration for doing my project;
And to all interested researchers.

Acknowledgements

My deep thanks and gratitude are due to Allah, the Almighty, who granted me knowledge. Without his support and guidance, this work would not have been possible, so all praise is to ALLAH.

Then, I would like to acknowledge and extend my heartfelt gratitude and appreciation to those whose kindness, patience and support were the candles that lightened my way toward success:

First, I would like to express my sense of gratitude and thanks to my supervisor Associate Professor Wesam Ashour for his support, advice, and encouragement throughout this study. I wish also to thank the discussion committee for their efforts.

I also wish to acknowledge my lecturers and academic staff in computer engineering department in The Islamic University of Gaza.

Finally, I reiterate my cordial acknowledgment and high appreciation to all those who helped in this study.

ملخص

أمن الشبكات واحدة من أخطر المشاكل في العالم وذلك بسبب الزيادة المستمرة للأنشطة الخبيثة واختراق الشبكات وتعطيلها. ومع إزدياد استخدام خدمات الويب في الكثير من الأنظمة على سبيل المثال: خدمات الحكومة الإلكترونية، الخدمات المصرفية، البريد الإلكتروني والتسويق الإلكتروني، باتت هذه الخدمات مهددة من قبل الهجمات الخبيثة.

تستخدم أنظمة كشف التسلل بشكل واسع لحماية أنظمه المعلومات والحد من الأضرار الناتجة جراء هذه الهجمات. وحيث أن العديد من الأنشطة الخبيثة والإختراقات لا تزال خفية عن تقنيات الحماية والأمن، أصبحت الحاجة إلى نهج فعال لكشف وتحديد مثل هذه الهجمات في إزدياد.

تحاول العديد من الدراسات إيجاد أفضل نموذج لأنظمة كشف التسلل لتحقيق أفضل معدل اكتشاف وأدنى معدل انذار كاذب. وقد استخدمت مختلف خوارزميات الذكاء الصناعي وتنقيب البيانات في هذا المجال مثل: خوارزميات التجميع، الشبكات العصبية، البيز البسيطة وشجرة القرارات، الخ ... ينقسم نظام كشف التسلل إلى نوعين رئيسين: أنظمة الكشف المعتمدة على أنماط البيانات وأنظمة كشف البيانات الشاذة. يستخدم النوع الأول للكشف عن هجمات معروفة عن طريق مقارنة أنماط البيانات بأنماط الهجمات المعروفة، أما النوع الثاني يستخدم للكشف عن البيانات التي تحيد عن سلوك البيانات الطبيعي.

تقترح هذه الدراسة خوارزمية تجميع جديدة للبيانات كبيرة الحجم، يمكن أن تولد مجموعات على النحو المحدد من قبل المستخدم. تستخدم خوارزمية التجميع الجديدة مزايا كلاً من خوارزمية تقارب الانتشار وخوارزمية تجميع المعكوس الوزني. التجارب على الخوارزمية المقترحة تبين أنه يمكنها أن تولد عدد محدد من المجموعات مباشرة دون أي ضبط مسبق، ويمكن تجميع البيانات كبيرة الحجم بكفاءة أكبر من الخوارزميات الأخرى ذات الصلة. وتشير النتائج إلى أن طريقة التجميع المقترحة يمكنها أن توفر الكثير من الوقت وتحقق نتائج تجميع أكثر فعالية ودقة. ثم تستخدم هذه الدراسة خوارزمية التجميع المقترحة لتقترح اثنين من نماذج كشف البيانات الشاذة الهجينة التي تحاول تحسين أداء نظام كشف التسلل. النموذج الأول هو نموذج هجين يعتمد على خوارزمية التجميع المقترحة وخوارزمية البيز البسيطة. تستخدم خوارزمية التجميع لتجميع كافة البيانات إلى مجموعات على أساس سلوكها مثل النشاطات الخبيثة وغير الخبيثة. في المرحلة الثانية، يتم استخدام البيز البسيطة لتصنيف البيانات الي الفئات الصحيحة. أما النموذج الثاني فهو نموذج هجين أيضا يجمع بين خوارزمية التجميع المقترحة وخوارزمية شجرة القرارات بدلاً من خوارزمية البيز البسيطة. استخدمت قاعدة بيانات KDD Cup '99 المتخصصة لتعليم النظام وتقييم أداء الأنظمة المقترحة؛ وقد قمنا في هذه الدراسة بعمل مقارنة بين النماذج المقترحة والنماذج الهجينة السابقة والنماذج التقليدية وتبين تفوق الأنظمة الهجينة المقترحة على النماذج والأنظمة السابقة؛ سواء الهجينة او التقليدية.

Abstract

Network security is one of the most serious problems in the world because of the continuing increase in malicious activities and networks attacks. The increasing use of web services in many systems such as e-government services, banking services, E-mail and e-commerce expose these services to several types of malicious attacks. Intrusion Detection Systems (IDS) are widely used to protect information systems and reduce the damage caused by these attacks. Some of the malicious activities are still hidden, and there is an urgent need to continue in developing new effective and adaptive approach to countermeasure such activities. Many studies try to find the best model for IDS to achieve the best detection rate and lowest false alarm rate. Various artificial intelligence and data mining algorithms have been used in this field such as Clustering algorithms, Neural Networks, Naïve Bayes, Decision Tree, etc. IDSs are divided into two main types: misuse detection and anomaly detection. The former is used to detect known attacks by extracting features from network traffic, matching them to a list of signatures, while the latter identifies any anomalous behavior by computing deviation from normal behavior. This study proposes a new clustering algorithm called IWC-KAP for large-scale data sets. IWC-KAP can directly generate K clusters, as specified by the user. It retains the advantages of K-Affinity Propagation and Inverse weighted clustering algorithm. Experiments on IWC-KAP show that it can generate K clusters directly without any parameter tuning, and can cluster large-scale data more efficiently than other related algorithms. Moreover, given a specified cluster number, results show that the proposed clustering method can significantly reduce the clustering time and produce better clustering result in a way that is more effective and accurate than AP, KAP, and HAP algorithms. Furthermore, the study used the IWC-KAP to propose two hybrid anomaly detection models to improve the performance of intrusion detection system in term of detection, accuracy, and false alarm rate. The first model combines IWC-KAP Clustering algorithm and Naïve Bayes algorithm. IWC-KAP uses to cluster all the data into clusters based on their behavior, such as malicious and non-malicious activities. In the second phase, Naïve Bayes classifier uses to classify clustered data into correct categories. The second model combines IWC-KAP algorithm and Decision Tree algorithm instead of Naïve Bayes classifier. KDD Cup '99 dataset is used for training and evaluating the performance of the proposed models.

Contents

List of Figures

List of Tables

List of Abbreviations

1R	One-R Classifier
ANN	Artificial Neural Networks
AP	Affinity Propagation
APP	Adaptive Affinity Propagation
BIRCH	Balanced Iterative Reducing and Clustering Using Hierarchies
DARPA	Defense Advanced Research Projects Agency
DoS	Denial of Service
DR	Detection Rates
DT	Decision Tree
EM	Expectation Maximization
FC	Fuzzy Clustering
FP	False Alarm
FSVM	Fuzzy Support Vector Machine
HAP	Hierarchical Affinity Propagation
HID	Host Based IDS
HMM	Hidden Markov Models
IDS	Intrusion Detection System
IWC	Inverse weighted clustering algorithm
KAP	K-Affinity Propagation
KDD	Knowledge Discovery and Data Mining
KM	K-means Clustering
K-NN	K-Nearest Neighbors
MEAP	Multi-exemplar Affinity Propagation
NB	Naïve Bayes
NIDS	Network Based IDS
NMI	Normalized Mutual Information
PLSSVM	Least Squares Support Vector Machine
Prob	Probing Attack
R2L	Remote to Local Attack
SMV	Support Vector Machine
TANN	Triangle Area-based Nearest Neighbors
U2R	User to Root Attack
WEKA	Waikato Environment for Knowledge Analysis

Chapter 1

Introduction

With the rapid growth of network technology and using it to transfer sensitive information, network security has become a critical problem. Maintaining the security of information is technically difficult and costs a lot of money.

Cyber-attacks are the big threat for our security. Cyber-attacks can move over networks and cause damage to data even for those who are browsing the Internet from home. Therefore, we must give cyber-attack detection the highest priority in order to secure our network. Unauthorized access to files and network, as well as other serious security threats can be detected through the use of intrusion detection systems (IDS).

In modern networks, an IDS is an essential component in network security infrastructure along with other components such as access control (ACLs), encryption tools, and firewalls. IDS identifies any activity that break the security policy of the different areas within the computer environment and network [1,2].

An IDS is able to send alerts once an attack is detected so that system administrators can respond and act accordingly. Thus, serious system damage can be reduced immediately [3].

While building an IDS, many issues must be taken into consideration. The issues include data collection, features extraction, intrusion recognition, reporting, and response. However, intrusion recognition is the core issue. Audit data are examined and compared with detection models, which classify the data to be normal or intrusion, so that both successful and unsuccessful intrusion attempts can be identified [2].

Data mining is the analysis of datasets to detect the underlying models from a set of training data and to optimize the data in novel ways that are both understandable and useful for the data owner [4]. In fact, the process of automatically constructing models from data for intrusion detection is not trivial because of the huge volume of network traffic, highly unbalanced attack class distribution, the difficulty to realize decision boundaries between normal and intrusion behavior, and requiring continuous adaptation to a constantly changing environment [2]. Data mining techniques can be used in IDS to classify network connections into normal or intrusion data based on labeled training data in misuse detection, and to group similar network connections together in clusters according to a given similarity measure in anomaly detection [5,6].

Hybrid classifier combines several techniques of data mining, in order to further improve the performance of the detection system [7].

In particular, there are two types of hybrid classifiers. The first type of hybrid classifiers is based on a combination of clustering techniques and classification techniques, such as K-means Clustering algorithm and Naive Bayes classifier [8]. The second type of hybrid classifier is based on combining various classifiers [9] such as Bayesian and K-NN classifiers.

In the model of the hybrid data mining techniques that combines clustering and classification techniques, clustering is used as the first module for pre-classification technique and classification is used later for the final classification task [10,11]. In particular, clustering can be used to as data reduction technique by filtering out data that is not representative. The data that was not clustered correctly can be considered as outlier's data. Typical data devoid of outliers apply to train the classifier in order to get better classification rate.

1.1 Security Fundamentals

In cyber security, asset, vulnerability, and attack are the fundamental components.

1.1.1 Assets

An IT asset is any company-owned data and information, System (applications, operating systems), Hardware, or Reputation that is used in the course of business activities [12,13].

1.1.2 Vulnerabilities

A vulnerability is a weakness in a system which allows an attacker to damage assets [13]. To exploit a vulnerability, an attacker must use at least one applicable to establish a connection with the system that comprise the weakness.

1.1.3 Threats

A threat is the actions that can be used to take advantage of the vulnerability and cause a negative impact on the network. The fact that the threat might occur means that those actions that could cause damage must be guarded against. Those actions are called attacks [14]. So Network attacks are defined as a set of malicious activities that are used to exploit vulnerabilities to damage, deny, or destroy service and information in computer networks assets. A network attack is executed through the data stream on networks and aims to compromise the Availability, Confidentiality or Integrity of the networks. The one who executes such activities, or cause them to be executed, are called attackers.

Threats can be categorized in many ways [15,17], an example of categorized ways is Microsoft's STRIDE threat model [16]. STRIDE categorize threats by the damage it causes to the assets, as following.

1. Spoofing identities: The attacker pretends to be somebody else.
2. Tampering with data: attacker alters data or settings to give him more privileges.
3. Repudiation: User denies making an attack, spending money.
4. Information disclosure: loss of information value by disclose to the wrong parties.
5. Denial of service (DoS): DoS attacks are preventing websites operation.
6. Elevation of privilege: elevation of privilege refers to illegitimately user gains privileges of the root user.

Another categorization is used in KDD Cup dataset, the training and testing data covers four major categories of attacks [15]:

1. Denial of Service attacks (DoS): is a class of attacks in which the attacker makes computing and memory resources too busy or too full to handle the legal requirements, thus denying legitimate users access to the machine [18].
2. Remote to Local (User) Attacks (R2L): is a class of attacks in which the attacker sends packets to a machine on the network, and then exploits the vulnerability in that machine to illegally acquire local access to the machine. This occurs when the attacker – who is able to send packets to the machine on the network, and does not have a legitimate account on that machine, exploits some vulnerability to acquire access equivalent to a local user on that machine [19].
3. User to Root Attacks (U2R): is a class of attacks in which the attacker starts with a normal local user access on the system and exploits further vulnerability(s) to gain root user access on the system. The attacker who has normal local user access on the system can conduct accumulated sniffing passwords, social engineering or dictionary attack to gain access to the system as a root [20].
4. Probing: is a class of attacks in which the attacker scans the network to find known vulnerabilities or to gain information. The attacker probes machines and services that are available on the network and use the collected information to exploit known vulnerabilities [21].

The information technology security involves the protection of information assets through the prevention, deletion or recovery of assets from security threats and vulnerabilities.

Security is described through the accomplishment of the basic security services confidentiality, integrity, and availability (also known as the CIA triad) to counter threats upon the systems [14].

Data Integrity service ensures that during their transmission the data is not altered by unauthorized principles.

- Confidentiality

  Confidentiality prevents wrong people from reaching sensitive information while making sure that the right people can actually get it [22]. In short, Confidentiality is roughly equivalent to privacy.

- Integrity

  Integrity involves maintaining the consistency, accuracy, and truthfulness of data or resources. It ensures data was not altered by unauthorized user whilst transmission. Integrity includes data integrity (the content of the information) and origin integrity (the source of the data, often called authentication) [14].

- Availability

  Availability refers to the ability to use the information or resource desired [14,12]. Availability reflects the reliability of the system as well as the system design because of the system that are not available at least as bad as no system at all [14]. The aspect of availability that is relevant to security is that someone may deliberately arrange to deny access to data or service by making it unavailable. An example of availability attack is the denial of service attacks.

Apart from the above three main properties, there are other properties which may be considered part of computer security. These include Authentication, Access control, Nonrepudiation, accountability, reliability, fault-tolerance and assurance

1.2 Challenges in Computer Security

Ideally, a computer system can be made perfectly secure if all the above mentioned properties were well satisfied. However, in reality a perfect security system does not exist [22]. Any system can be subjected to break of integrity, confidentiality, and availability. Hereby, it declares itself in an insecure situation. To address such issue, it is valid to assume that any system will be breached at some point, and therefore, detection mechanisms should be placed as part of an overall protection mechanisms [22].

- Protection

  The proactive part of security consists of protecting the asset. The asset must be protected to counter any break of integrity, confidentiality or availability. [12].

- Detection

  Unless perfect security system cannot be achieved, it is expected that protection measures might not be able to protect assets under all cases. Therefore, appropriate detection measures are required. These measures are used to detect a potential break of security and their efficacy depends on the time taken to detect. Time of detection may differ based on the difference in assets and the value of the asset. Another factor that contributes to the efficiency of a detection mechanism is the number of false alarms it generates. A false alarm may be a false positive or a false negative. The higher the number of false alarms, the slower and more expensive the detection process is [12].

- Response

  Detection process can be extended to include response to a breach. The response type differs from situation to another and would depend on the exact security requirement. Common response model includes evaluating the damage, recovering from the damage, improving with experience, etc [12].

1.3 Intrusion Detection

Intrusion detection is a type of security management system that attempt to validate the integrity, availability or confidentiality of data on computers and networks [12].

An IDS is a hardware and / or software system that is designed to detect the violation of a security policy or attempts to disable the system.

These violations may be caused by external people from outside an organization (i.e. attackers) or by employees inside the same organization (i.e. insiders). Although the progress has been made to detect violations by outsiders and insiders, violations are still difficult to detect. [12]

IDS functions by monitoring all network activities in an attempt to detect known or unknown attacks [23]. The main objective of IDS is to alarm the system administrator if any suspicious activities take place [24]. Fig 1.1 illustrates the components of an IDS.

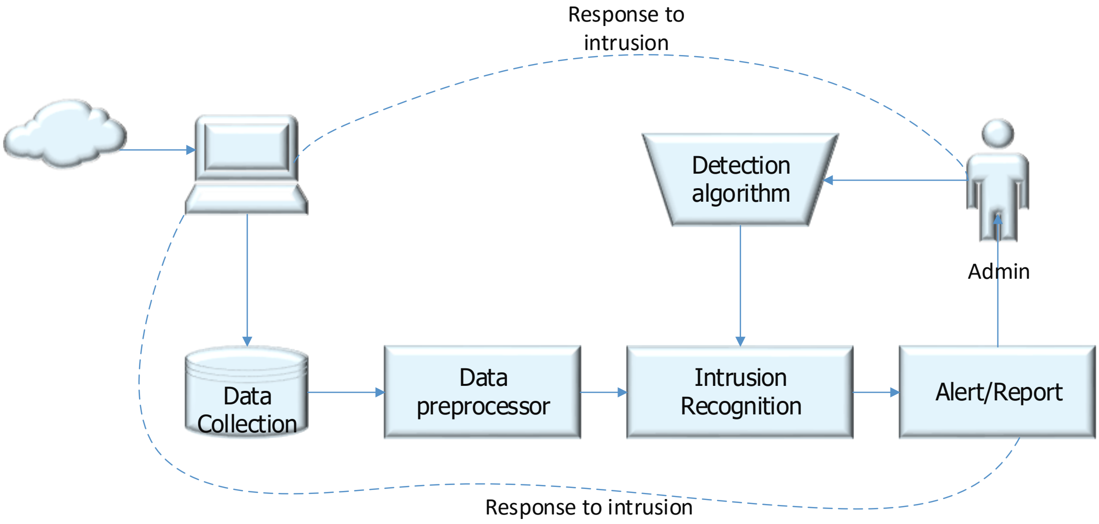

Figure 1.1 Overview of IDS

The IDS performs its task according to classification and determination rules. The rules are split into two main parts: a pre-existing knowledgebase and a set of classification algorithms. The pre-existing knowledgebase essentially includes (1) a set of computed parameters based on the past network traffic, (2) a set of known and labeled attacks events, and (3) a group of data sources. The classification algorithms are essentially build based on either Artificial Intelligence, statistical modeling, or a hybrid of both. The performance of intrusion detection systems is affected by different factors such as the quality of the pre-existing knowledgebase, the robustness of the classification algorithms, and the uniqueness of attacks or anomalous events. The pre-existing knowledgebase should be updated regularly so that any recently discovered malicious codes or attacks can be identified and the classification parameters update accordingly.

IDS is intended to detect as many types of attacks as possible regardless of the whether the source of an attack was from inside or outside an organization. Furthermore, detection should take place with least number of false alarms in the best possible time [14].

1.3.1 Motivations behind Intrusion Detection

Intrusion Detection has received important motivation due to the following reasons: [12,25]

1. If an intrusion is detected instantly, an attack can be identified quickly and ejected from the system before any data are compromised or any damage is done. Even if the detection is not sufficiently timely to preempt the attacker, the sooner that the intrusion is detected, the less the amount of damage done is and more recovery can be performed quickly.
2. A powerful intrusion detection system can work as a deterrent. Alerts can be sent to by IDS to network administrator to take the proper action. Additionally, an IDS is considered the heart of intrusion prevent systems.
3. Intrusion detection allows the collection of information about intrusion methods that can be used to increase the intrusion prevention facility.

1.3.2 Goals of Intrusion Detection

Along with the motivations, the goals of intrusion detection can be summarized as below [12,14]:

- Detect several type of intrusions as possible. Insider intrusions, as well as outsider intrusions, are of interest. Furthermore, both known and unknown attacks should be detected.
- Detect intrusions opportunely. "Opportunely" here need not be in real time. Often, it suffices to detect an intrusion in a short period of time. Real-time intrusion detection raises subjects of responsiveness. If every command and action must be analyzed before it can be executed, only a very simple analysis can be done before the computer (or network) being monitored becomes unusable.
- Present the analysis in an easy to understand and simple format.
- Detect as much as possible intrusions thereby reduce the number of false alarms.

1.3.3 Types of Intrusion Detection

IDS can be identified by two techniques, anomaly detection and signature-based detection (i.e. misuse) [26,27].

Based on their functionality (classification rule), these techniques can be classified into misuse and anomaly detection.

A misuse detection technique typically can detect known attack by extracting features from network traffic, matching them to a list of signatures, as the case in antivirus application. When data is passed to the IDS, it applies the rule set in the signature database to the data to determine if any sequences of data match any of the rules. If so, it reports that as a possible intrusion in underway. Misuse-based intrusion detection systems often use expert systems to analyze the data and apply the rule set [14].

The weakness of the misuse-based IDS approach is that it fails to identify slow attacks that extend over a long period of time. To detect these types of attacks, huge amounts of information must be held for long time periods. Another issue is that this type of IDS requires updating of the signature database with new signatures data frequently, otherwise, it fails to detect unknown attacks if the signature is not in the database [28].

To maximize the advantages of misuse and anomaly detection techniques and avoid their disadvantages, there are a lot of proposed hybrid approaches in the last years such as [28,29]. Fig 1.2 [31] shows a misuse detection system flowchart.

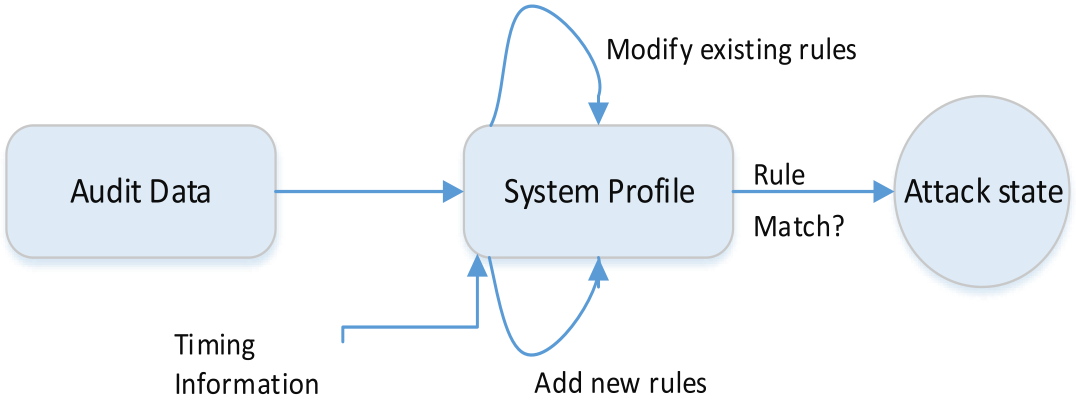

Figure 1.2 Misuse Detection Systems [30]

Anomaly-based detection is designed to identify any anomalous behavior by computing deviation from normal behavior [31,32]. In this type, observable behaviors of a system are used to build models for normal system operation. These behaviors may include audit logs, network sensors, system calls, etc. [12].

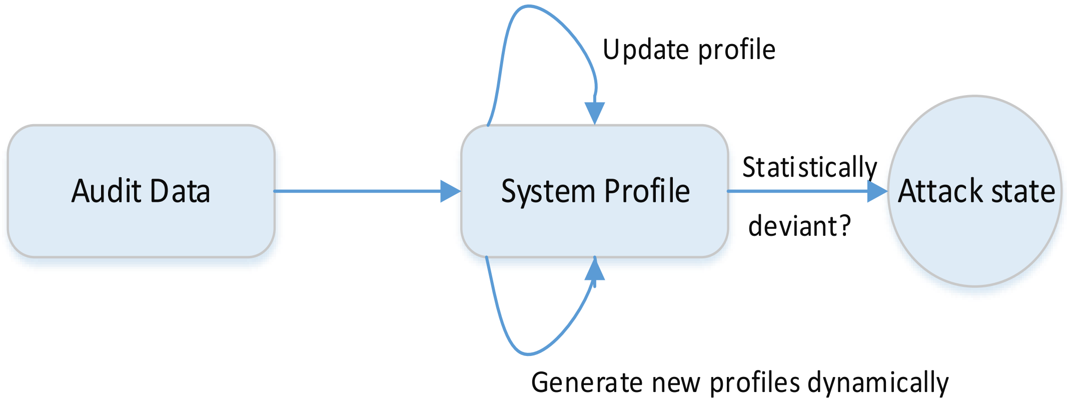

Figure 1.3 Anomaly Detection Systems [30]

There are two assumptions that the attack behavior is rare and it is different from the normal behavior [2,33]. However, these assumptions are not always true because of the high degree of similarity between some kinds of normal and intrusions connections, this leads to generate large quantities of false alarms. [34]. Fig 1.3 [30] shows Anomaly detection system flowchart.

The main advantage of this type is that it can detect novel attacks. The disadvantages of that system are the high false alarm rate and the practical difficulty in defining normal behavior pattern [32,35], and the behavior of the users on a networked system might not be consistent enough to effectively implement that IDS [13]. However, its major difficulty lies in discovering boundaries between normal and attack behavior, due to the deficiency of attack samples in the training phase [2].

The Anomaly-based detection is identified by three different models; threshold-based, profile-based, and Markov model [38,14]. The first model uses a threshold metric. This model measures the frequency of anomalous events in a specified period [32]. The count of events that occur over a period of time determines the alarm be raised if fewer then m or more than n events occur

The second model is profile-based, this model focuses on the analysis of the current or the historical user behavior and detects any outlier values based on a series of measures as standard deviation, median, mean, or interval estimates of various user activity-related parameters and variables (e.g., the types of protocol, the connection statuses, the login time zone, and the length of connections) [32]. This model provides more flexibility than the threshold model. Administrators can tune it to distinguish better than the threshold model. But complexity comes at the expense of flexibility. In particular, an explicit assumption is that the behavior of processes, and users, can be statistically modeled. If this behavior matches a statistical distribution (such as normal distribution or Gaussian), determining the parameters requires experimental data that can be obtained from the system. But if not, the analysts require other techniques such as clustering to determine the characteristics and the values that indicate anomaly behavior [14].

The third model is a Markov model. It examines the system at any given time. Events leading up to that time had put the system in a particular state. When the next event occurs, the system moves to a new state. Over time, a set of transition probabilities can be developed. When the event that causes the lowest probability transition happen, the event is considered anomalous. This model suggests that the concept of "state" or past history, can be used to detect anomalies. Now the anomalies are no longer dependent on the statistics of the occurrence of individual events but it depends on sequences of events [12].

IDS can also be classified according to protected system to host-based (HIDS), network-based (NIDS), and Hybrid system(HID). The main function of HIDS is internal monitoring (within a computer or machine). HIDS detect malicious activity and warn administrators correspondingly. NIDS is used to monitor and analyze network traffic to protect a system from network-based threats. A NIDS operate on network segments and analyze that segment's traffic. When threats are discovered, based on its severity, the system can take action such as notifying administrators. HID use a combination of both HIDS and NIDS, so it provides more flexibility and security to a system [37].

1.4 Thesis Contribution

Various data mining techniques were proposed in recent years for building IDS. The proposed techniques were trying to find the optimal accuracy, detection, and false alarm rate.

This research proposes a new clustering algorithm called IWC-KAP for large-scale data sets. IWC-KAP can directly generate K clusters as specified by the user. It retains the advantages of K-Affinity Propagation (KAP) [38] and Inverse weighted clustering algorithm IWC [39]. IWC-KAP suggests two mechanisms for splitting data into subsets and then applies the KAP algorithm on each subset to find the local exemplars. IWC algorithm is applied on all local exemplars to find a specific number of global exemplars; then all the data points are re-clustered into new clusters by the global exemplars. Experiments on IWC-KAP show that it can generate K clusters directly without any parameter tuning, and can cluster large-scale data more efficiently than other related algorithms and reduce the clustering time.

Then the research proposes two hybrid anomaly detection contributions depend on the IWC-KAP. First a hybrid learning approach based on the combination of IWC-KAP Clustering with Naïve Bayes Classifier (NB) [40] or Naïve Bayes Multinomial (NB Multinomial) [41]. IWC-KAP Clustering is used to cluster all data into groups based on its behaviour such as malicious and non-malicious activity. In the second stage, Naïve Bayes Classifier is used to classify clustered data into correct categories. The second model is hybrid IDS using a new combination of IWC-KAP Clustering, with Decision Tree Classifier (DT) [42] instead of NB. The improvement that the proposed models achieve is due to combining clustering and classification together. Clustering has significant advantages over classification techniques and helps to identify the group of data that behave similarly. The proposed approached shows significant improvement in detection rate accuracy, and false alarm rate compared to other approaches.

The proposed models in this study are evaluated over a real network connections data which are generated from the Defense Advanced Research Projects Agency (DARPA) network connections. Data were prepared by ACM Special Interest Group on Knowledge Discovery and Data Mining in the Knowledge Discovery and Data Mining 1999 (KDD Cup ‘99) contest. KDD Cup dataset suffers from some problems [15], KDD Cup is imbalanced and contains redundant connections and this cause the learning algorithms and the evaluation results to be biased towards the frequent records. We evaluate the proposed models using 10% KDD Cup (a reduced version) dataset.

1.5 Thesis Organization

The study is organized as follows. Chapter 2 discusses related work and theoretical background that make use of artificial intelligence algorithms in the field of IDS. Chapter 3 presents the proposed clustering and IDS models. Chapter 4 shows and discusses the experimental results. Finally, Chapter 5 summarizes the study and discusses future work.

Chapter 2
Background

This chapter takes a fast review to previous works in the field of IDS and discusses a background about artificial intelligence algorithms used in this study.

2.1 Literature Review

Most of artificial intelligence and data mining techniques has been introduced to develop intrusion detection systems, researchers in this field concentrating in using clustering algorithm or one of its improved versions and apply it to IDS, recent researches uses combinations models of artificial intelligence and data mining techniques to get improvement in performance (detection rate, accuracy, and false alarm). Firstly, we present samples of previous work of Affinity Propagation Clustering models then we present samples of previous work for hybrid IDS models.

2.1.1 Affinity Propagation Clustering

Affinity Propagation clustering algorithm [43,45] is based on passing messages between data points. Each data point receives availability a(i,j) message from the exemplars (members of the input set that are representative of clusters) and sends a responsibility r(i,j) to the example. The AP messages take into account different kind of competition. The availability message that is sent from candidate exemplar j to point i, reflects the accumulated evidence of how close point i to point j, while taking into account that point j may be an exemplar for other points. The responsibility message that is sent from data point i to candidate exemplar point j, reflects the accumulated evidence of how well-suited data point j is to serve as the exemplar for data point i, and taking into consideration other possible exemplars for point i. The input of the AP is a matrix of similarities between pairs of data points S= (s(i,j)), and the output is cluster exemplars of all data points and relationships between each point and its cluster's exemplar. The similarity function s(i,j) in the similarity matrix indicates how well the data point j is suitable to be the exemplar of a data point with index i. The diagonal element of the similarity matrix S(k,k) indicates the 'preference' of data point with index k, so exemplars are that data points with larger values of S(k,k).

The steps of the AP algorithm are as shown in Algorithm 2.1.

Algorithm 2.1 AP algorithm steps

```
1: Initialization the availabilities matrix to zero $a(i,k) = 0$
    $k$ is the number of exemplars
   $i \in \{1,2,\dots\dots,n\}$
  $n$ is the number of data points
```

```
2: Update the responsibilities by the following equation.
    $r(i,k) = s(i,k) + \max_{k\prime \neq k}\{a(i,k') + s(i,k')\}$
    where s is the similarity matrix between the data points.
3: Update the availabilities matrix
    $a(i,k) = \min\{0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}\}$
    Update self-availability by
    $a(k,k) = \sum_{i' \notin \{i,k\}} \max\{0, r(i',k)\}$
4: Find $sum = a(i,k) + r(i,k)$ for each point $i$ and the exemplar $k$ that maximize the sum.
5: For fixed number of iterations If exemplars do not change go to step (6) else go to Step (1)
6: Assign the data points to its exemplars based on the maximum similarity to find clusters
```

Multi-exemplar Affinity Propagation (MEAP) [46] proposed an extension of the single-exemplar model to a multi-exemplar one. MEAP can identify exemplars and a superexemplar for each cluster automatically. Each data point assigned to the most suitable exemplar and each exemplar assigned to the most suitable superexemplar. The superexemplar is defined as an exemplar that best represents the exemplars belonging to the corresponding cluster. The objective of the MEAP is to maximize the sum of all similarities between data points and the corresponding exemplars, plus the sum of all linkages between exemplars and the corresponding superexemplars. However, if the cluster number is prior knowledge, MEAP would not be able to make use of such knowledge directly in its learning process. Instead, it has to rely on re-running the process as many times as it takes by tuning parameters until it generates the desired number of clusters. It also consumes a large amount of time and memory while processing large-scale data.

Adaptive Affinity Propagation (AAP) [47] was proposed as a model to overcome the drawback of AP that is related to knowing the value of the parameter preference, and tries to produce an optimal clustering solution. AAP firstly finds out a range of preference, then searches the space of preference to find a good value, which can optimize the clustering result.

KAP [38] was modified to generate a given number of an optimal set of exemplars through Affinity Propagation. KAP can generate K clusters as the user specifies by adding one constraint in the process of message passing to confine the number of clusters to be K while keeping all AP advantages in clustering. Another advantage of KAP over AP is the confidence in one data item to be an exemplar is automatically self-adapted by KAP while the confidence in AP is a parameter specified by a user. Moreover, the computational cost overhead compared to AP is negligible. However, the limitations of clustering large-scale data are still existing as in AP. It still consumes time and memory while processing large-scale data.

Hierarchical Affinity Propagation (HAP) was the first algorithm to use AP algorithm on large-scale data [48]. HAP proposed an improved hierarchical AP clustering algorithm. The algorithm achieves efficient, accurate and no predefined parameter large-scale data clustering by applying hierarchical selection and partitioned clustering. In a hierarchical selection, AP algorithm is executed for each subgroup according to; firstly, finding well suited local exemplars for clusters in each subgroup. Secondly, AP is executed on all the local exemplars to find the global exemplars for all the data. In partitioned clustering, all of the data points are partitioned once again into new clusters by the global exemplars. One of the drawbacks of HAP is that when the number of clusters K is available, HAP, just like AP, cannot generate specified number of clusters directly. The second drawback of HAP is the time and memory consumption that comes as a result of using AP in finding the global exemplars.

2.1.2 hybrid IDS

Bouzida et al [49] compare the Decision Tree Classifier with and without principal component analysis, a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. They reduced computation time on KDD ‘99 dataset by a factor of approximately thirty, with a slight loss of overall accuracy from 92.60% to 92.05%.

Bouzida et al [50] concluded that while Neural Network is very interesting for generalization and very bad for detection of new attacks, Decision Tree Classifier have proven efficiency in both new attacks detection and generalization.

Tsai et al [51] proposed a hybrid learning model based on the Triangle Area-based Nearest Neighbors (TANN) which consists of K-means Clustering and K-Nearest Neighbors (K-NN) Classifier to effectively detect attacks. Initially, K-means Clustering is performed to cluster the training data to a cluster that represents one particular category of attacks, and then the K-NN Classifier is applied. Even though the proposed model has a high detection rate at 98.95%, but it came with a high false alarm rate at 3.83%.

Yassin et al [52] proposed a hybrid Intrusion Detection learning model based on K-means Clustering and One-R Classifier (KM+1R). The main solution is to break up the instances between the normal data and the attacks with a first step in a different cluster. Then, the clusters are variant into DoS, R2L, U2R, Probe attacks, or Normal. The KM+1R performance was measured using KDD Cup '99 datasets. KM+1R hybrid approach ascertains accuracy rate 99.26% and detection rate at 99.33%, but it still has a low false alarm rate at 2.73%.

Yassin et al [53] proposed hybrid Intrusion Detection learning model based on K-means Clustering and Naïve Bayes Classifier (KM+NB). They used the ISCX 2012 dataset to evaluate the performance of the proposed model. They concluded that the KM+NB had highly enhanced the accuracy rate by 99% and detection rate at 98.8%, while decreased the false alarm rate to 2.2%.

Purohit et al [54] proposed a hybrid Intrusion Detection learning model based on K-means Clustering, Naïve Bayes, and Decision Table majority approaches to improve the accuracy rate, detection rate, and false alarm. K-means clustering perform as a pre-classification to cluster a similar behavior of data in a single cluster. Next, the Naïve Bayes Classifier classified the clustered data into normal and abnormal classes to reduce the amount of misclassified results during the clustering stage. Then, the classified data pass into Decision Table majority for the successful progression. This model evaluated with KDD Cup '99 dataset. However, no result reported on this work.

Golmah [55] proposed an efficient hybrid Intrusion Detection learning model based on C5.0 Decision Tree and SVM Classifier. This model used DARPA dataset in the evaluation. This model achieves better performance compared to the individual SVM.

Kim et al [56] proposed a hybrid Intrusion Detection learning model hierarchically integrates a misuse detection and anomaly detection in a decomposed structure. The C4.5 Decision Tree Classifier used to build misuse detection model. This model used to decompose the normal training data into smaller subsets. The one-class SVM is used to create anomaly detection for the decomposed region. C4.5 Decision Tree does not form a cluster, which can degrade the profiling ability.

Khosronejad et al [28] proposed a hybrid Intrusion Detection learning model for anomaly detection based on Hidden Markov Models and C5.0 Classifier. This model achieves better accuracy in comparison to the HMM and reduces the limitations of HMM algorithm.

Ghanem et al [57] proposed a hybrid Intrusion Detection learning model based on multi-start metaheuristic method and genetic algorithm. This model used for anomaly detection in large-scale datasets. It has taken inspiration of negative selection based detector generation. This approach achieves a better accuracy in generating a suitable number of detectors compared to the other approaches as J48 Decision Tree, Naïve Bayes, Bayes Network, Bayesian Logistic Regression, Multilayer Feedback Neural Network, and Radial Basis Function Network.

Muniyandi et al [58] proposed an efficient hybrid Intrusion Detection learning model based on K-means and C4.5 Decision Tree Classifier. In this approach, initially K-means Clustering used to partition the training dataset into clusters. Then, C4.5

Classifier used to build the Decision Tree for each cluster. The C4.5 Classifier created rules that used to detect intrusion events. The test phase is implemented in two stages. In the first stage, finding the closest cluster for each instance using K-means. In the second stage, the Decision Tree corresponding to the closest cluster is selected to detect the class of the instance. In this work, K-means still have some shortcomings, such as the clustering output mainly depends on the selection of initial class centers. Also, clustering result does not include all class instances possibilities. This model has the accuracy rate at 90.17% and detection rate at 81.77%.

However, Al-Yaseen et al [59] proposed an efficient hybrid Intrusion Detection learning model based on modified K-means and the C4.5 Decision Tree Classifier. In this approach, the work depends on the proposed in [30] with a modification for K-means that adopted for choosing the initial centroids of clusters. This model has the accuracy rate at 90.22% and detection rate at 83.94%.

Tsai et al [60] introduced K-Means Clustering to cluster data instances into k-clusters. Then, a new dataset, which consists of the centers of clusters, is trained using Support Vector Machine (SVM). This approach gives high accuracy rate and high detection rate for almost all types of attacks, but it also produces high false alarm rate.

Gang et al [61] proposed a new approach to intrusion detection using Artificial Neural Networks and Fuzzy Clustering (FC-ANN). Fuzzy Clustering is applied to generate different subsets before being trained in different ANN models to develop different models. After that, a fuzzy aggregation module is used for result aggregation. Each subset of the training set has less complexity through the use of Fuzzy Clustering. This allows ANN to learn each subset more deeply to detect low-frequency attacks such as R2L and U2R attacks. However, compared to Naïve Bayes Classifier, this approach led to a lower detection rate in Probe attacks.

Tsuruokaand et al [52] proposed an approach that combined the Naïve Bayes Classifier with well-established Expectation Maximization (EM) algorithm to exploit the unlabeled data. That approach introduced a class distribution constraint on the process of EM algorithm. This constraint maintains the class distribution of the unlabeled data consistent with the true distribution of the labeled data. Hence, it prevents EM algorithm convergence to an undesirable state.

Farid et al [62] proposed a new adaptive network intrusion detection algorithm that use Naïve Bayes Classifier and Decision Tree. The proposed algorithm reduces false positives and performs high detection rates for different types of network intrusions using limited computational resources.

Cao et al [63] proposed combining Radial Basis Function Neural Network proposed in [61] and Artificial Immune Network algorithm. This paper, firstly employed multiple granularities artificial immune network algorithm to get a hidden neuron candidate. Then a cosine Radial Basis Function neural network is trained based on gradient descent learning process, achieving accuracy ability and significant pattern classification. Experimental results indicate that the proposed approach performs reasonable detection rate, but could be improved.

Shaohua et al [64] proposed intrusion detection based on Fuzzy SVMs (FSVM) to improve the classification accuracy. The aim of the clustering algorithm is to construct a new training set using clusters centers and then using FSVM to obtain a support vector from the new sets. Although experimental results indicate that the proposed method has increased the accuracy rate, yet the accuracy rate is not acceptable.

Amiri et al [65] used a feature selection method to delete unimportant features to improve existing classifiers' performance that have heavy computational challenges for large datasets. Thus, this work introduced an improved Least Squares Support Vector Machine (PLSSVM). PLSSVM has a good result in classifying Normal and Probe, but a large number of dynamic attacks such as DOS and U2R were not detected because their behavior is very similar to normal behavior.

Horng et al [66] proposed SVM-based IDS with preprocessing phase using Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) as feature selection procedure to delete the unimportant features. BIRCH algorithm improves the performance of SVM while simple feature selection procedure enabled SVM model to classify some data correctly. Although this method has a good result in some data, but it could not make a significant difference between Normal and R2L data. The prediction rate of this category dropped dramatically.

Huy Anh et al [67] proposed an evaluation of a comprehensive set of classifiers to detect the four attacks that existed in the KDD data set. The best classifier for each attack and two appropriate classifiers are proposed for their selected models. Nevertheless, it can improve the detection rate of R2L attacks.

Meera et al [68] proposed the best classifier for each category of attacks by the evaluation of a wide range of different classifiers using the KDD dataset. However, there is no false alarm and detection rate reported by the author.

Muda et al [8] proposed a hybrid learning approach that combined K-Means clustering and Naïve Bayes classifier to improve accuracy and detection rate. The proposed approach clusters similar data instances based on their features by using a K-Means clustering. Next, it uses Naïve Bayes classifier to classify the clustering result into attack classes as a final classification task. Experimental results indicate that the proposed approach performs reasonable detection rate, but could be improved.

In short, various techniques have been proposed in the field of intrusion detection, but there is still room to improve detection rate and accuracy, and reducing false alarm rate.

2.2 Decision Tree

The Decision Tree is a classification technique where each dataset attribute is tried to classify in user-defined classes or types [42].

The input attributes of the decision tree can be continuous or discrete, depending on the needs of the user, also the output value continuous or discrete. For example, someone may divide network packet whether it is only attack or normal without considering attack type, while other may divide it with respect to attack types then the output will be specific attack type and normal. With considering whether the packet is an attack or not the output will be a Boolean value like TRUE/FALSE. Decision Tree is easy to understand and visualize, making its explanation for the resulting value easily explained by Boolean logic. [69]

A simple way to construct decision tree as shown in Fig 2.1. The Decision Tree is defined [40] using Decision Tree learning algorithm. The major step includes choosing an attribute from the attributes, adding the best attribute at each level of the tree. A tree begins with the question "which is the attribute should be tested at the root of the tree?". The choosing attribute is totally depending on how that attribute able to do the classification of examples. A perfect attribute is the one that divides the examples in the user-specified classes. A useless attribute leaves the example sets with roughly the same proportion of specified classes. Here every internal is acts as a test. First best attribute acts as the root node of a tree. A descendant of the root node is then created by using best attribute for a dividing dataset of the appropriate descendant node. The entire process is then repeated so that all examples get classified as per need.

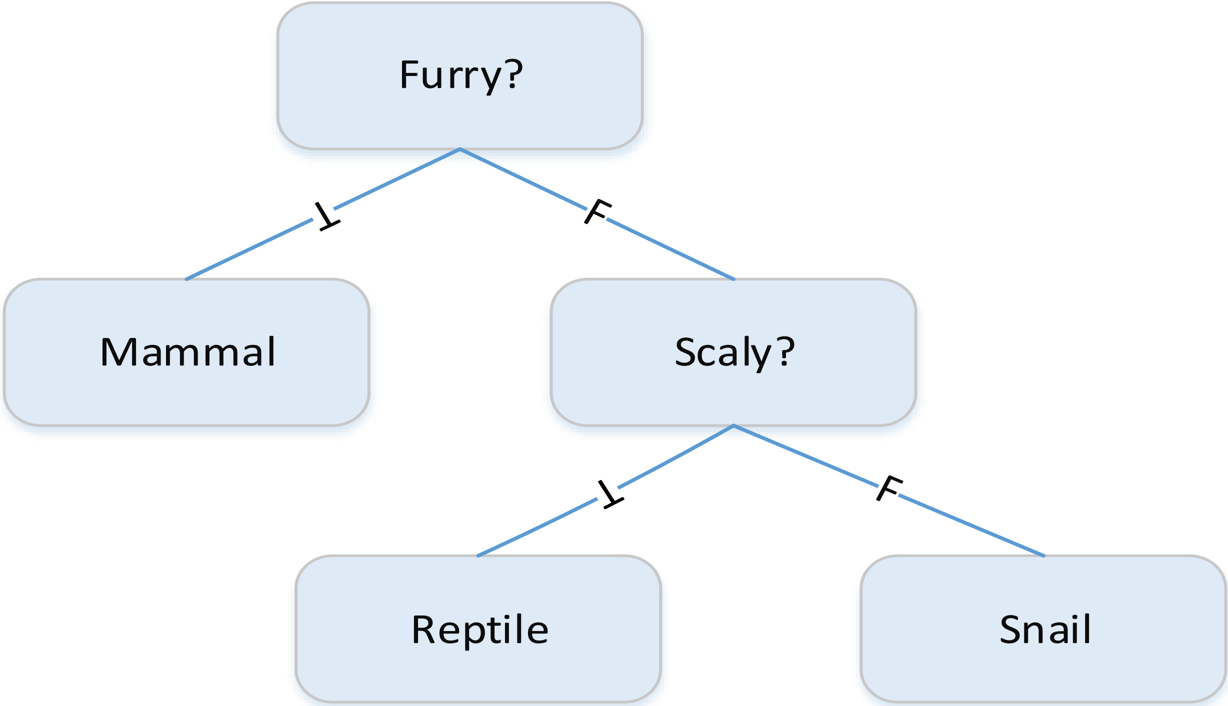

Figure 2.1 Decision tree diagram

2.2.1 J48 Decision Tree

J48 is a Java implementation of the C4.5 algorithm in the WEKA which is an open source data mining tool [70]

J48 is one of the most famous classification algorithms in data mining. It operates in a divide and conquer manner, which partitions recursively all training data based on its attributes as the stop conditions are met. [71,72] The J48 consists of nodes, edges, and leaves. Each J48 node has its corresponding data set; this specifies the attribute for better split the data set into its classes. Each node has several edges that specify value ranges or possible values of the selected attributes on the node. According to the specifications of the edges the node's data set is divided into subsets, and for each data subset the J48 creates a child node and repeats the dividing process. When the node follows the stopping rules because no future distinguishing attribute can be determined, or it contains homogeneous data sets, J48 ends the demarcation process and the node is labeled as follows the class name of the data set. [71] The labeled node is called as a leave node. In this way, the J48 partitions the training data set recursively and creates a tree-like structure.

The primary decision tree algorithms issue is to locate the attribute that best splits the data set into corresponding classes. J48 generates decision trees by training data sets using the information entropy concept. In other words; it is based on the highest gain of each attribute. Information gain is calculated using:

$$IG(S,A) = Entropy(S) - \sum_{i=1}^{n} \frac{Si}{S} \times Entropy(Si) \qquad (2.1)$$

Where IG (S, A) is the information gain of set S after a split over the A attribute, Entropy(S) is the set S information entropy, n is the number of different values of attribute A in S, A is the proportion of items possessing Ai is the value for A in S, Ai is the i th possible value of A, and Si is a subset of S containing all items where the value of attribute A is Ai. The entropy is obtained as follows:

$$Entropy(j) = \sum_{j=1}^{n} fs(j) - \log_2 fs(j) \qquad (2.2)$$

Where fs(j) is the proportion of the value j in the set S, n is the number of different values of the feature in S (entropy is computed for one chosen attribute). After creating the tree by maximizing the gain, J48 decomposes the data space such that individual decomposed regions become homogeneous. Then, J48 performs the final pruning step. This action reduces the misclassification caused by specializations throughout training set. Thus, it makes the tree more general.

2.2.2 J48-Graft algorithm

J48-Graft algorithm generates a grafted decision tree from the J48 algorithm. Unlike the J48, the grafting technique is an inductive process that adds nodes to inferred decision trees to reduce prediction errors. The J48-Graft algorithm classifies region of the multidimensional space of attributes not occupied by the training examples [73]. This process is often shown to improve predictive accuracy. A special analysis could suggest that the decision tree grafting is the direct reverse of pruning. Instead, it is claimed that the two processes are complementary. This is because, as the standard tree growing techniques, pruning uses only local information, whereas grafting uses non-local information. The use of both grafting and pruning together is demonstrated to provide the best overall predictive accuracy over a representative selection of learning tasks [74].

2.3 Naïve Bayes and Multinomial Naïve Bayes

Naïve Bayes Classifier [40] is a simple constructing classifiers technique. Naïve Bayes Classifier assigns class labels to feature vector values. The class labels are drawn from a limited range. It is not a unique algorithm for training such classifiers, but a family of algorithms based on a common principle [75]. For some types of models of possibilities, Naïve Bayes Classifiers can be trained with high efficiency with supervised learning setting. In many practical applications, maximum likelihood method used for parameter estimation in Naïve Bayes models. In other words, you can work with Naïve Bayes models without accepting or using any Bayesian method.

Naïve Bayes Classifiers advantage is that it requires only a small amount of training data to estimate the classification parameters.

All Naïve Bayes Classifiers assume that each of the features it uses are conditionally independent of the value of any other feature. More formally, for calculating the probability of observing features f_1 through f_n given some class c, under the Naïve Bayes assumption the following holds:

$$p(f1,...,fn|c)=\prod_{i=1}^{n} p(fi|c) \qquad (2.3)$$

This means that by using a Naïve Bayes model to classify a new example, the posterior probability is much simpler to work with:

$$p(c|f_1,...,f_n)\propto p(c)p(f_1|c)...p(f_n|c) \qquad (2.4)$$

Of course, these assumptions of independence are rarely true. However, practice Naïve Bayes models performed surprisingly, even on complex tasks where it is clear that strong independence assumptions are false.

So far we have said nothing about the distribution of each function. In other words, what is left $p(f_i|c)$ undefined. In the Multinomial Naïve Bayes Classifier [41] each $p(fi|c)$ is a multinomial distribution, rather than some other distribution. This works well for data that can easily be turned into counts.

In summary, the difference between Naïve Bayes Classifier and Multinomial Naïve Bayes Classifier is that, Naïve Bayes Classifier is a general term that refers to conditional independence of each of the features in the model, while Multinomial Naïve Bayes Classifier is a specific instance of a Naïve Bayes Classifier that uses a multinomial distribution for each of the features.

Chapter 3

Proposed Frameworks

In this chapter, we present the contributions. The details are shown in the rest of this chapter.

3.1 Clustering Proposed Frameworks

3.1.1 The basic idea of the IWC-KAP clustering algorithm

The IWC-KAP clustering algorithm depends on KAP Clustering algorithm to achieve efficient and accurate clustering result for large-scale data where the number of clusters is known. The basic idea of the algorithm is that, data set is divided into several subsets, each of which can be efficiently clustered by the KAP algorithm. The resulting subsets exemplars are clustered through the proposed IWC algorithm [39] to get a specific number of global exemplars, and then each data point is clustered to its exemplar. This process is divided into four steps:

Step 1: data partition

The entire data points of the data set are split into several small subsets that can be efficiently clustered using KAP. Two methods have been used in this step to divide the data. In the first method, data is divided randomly into n subsets, in the second method, data is divided using the K-means algorithm. Through this step, the KAP clustering algorithm is directly applied on a large-scale data set.

Step 2: using KAP algorithm

The KAP algorithm is executed on each subset to select well-suited specific number of exemplars for each subset. The selection depends on the known clusters number of the data set and leads to obtaining the local optimal exemplars.

Step 3: find global exemplars and grouping data

IWC algorithm is used to find cluster centers from the all-local optimal exemplars. The selected cluster centers of the entire data set are called global exemplars.

Step4:

Each data point is grouped into its cluster by finding its global exemplar using similarities between each data point and all global exemplars as in K-means clustering. Each data point will fit into its cluster as indicated by its maximal similarity.

The above-mentioned steps are well-described into the following two algorithms. In the algorithm, 3.1 data is divided randomly into n subsets, in the second method, while in the algorithm, 3.2 data is divided using the K-means algorithm.

Algorithm 3.1

```
1: The data set $D$ divided into $k$ partitions randomly, denoted as $D1, D2, \ldots, Dk$
   where $k$ is the number of partitions
   $D1 \cap D2 \cap \ldots \cap Dk = \emptyset$
   $D1 \cup D2 \cup \ldots \cup Dk = D$
2: For each partition $Di$, the KAP algorithm is performed to select the $n$ number of local exemplar set of this partition, denoted as $Ei$. Where $n$ is the number of clusters
3: Exemplars of all the partitions create a new data set, denoted as:
   $E = E1 \cup E2 \cup \ldots \cup Ek.$
   IWC will be used on the data set E to select the global exemplars of the entire data set, denoted as $Eg1, Eg2, \ldots, Egn$. Each exemplar $Egi$ ($1 \leq i \leq n$) will be regarded as a centroid of cluster $Ci$, which is $Ci = \{ Egi \}$.
4: For each point di in data set $D$ the similarities between $di$ and each exemplar $Egi$, denoted as sim($di$; $Egi$), are compared to find the exemplar point m with the maximal similarity.
   $m = \max_{j} sim(di; cj)$
   $then\ Cm = Cm \cup \{di\}$
5: Return the clustering result
   $D = C1 \cup C2 \cup \ldots \cup Cn$
```

Algorithm 3.2

```
1: The data set $D$ divided into $k$ partitions using K-means algorithm to cluster data to initial $k$ clusters, $k$ clusters denoted as $D1, D2, \ldots, Dk$
   where $k$ is the number of partitions
   $D1 \cap D2 \cap \ldots \cap Dk = \emptyset$
   $D1 \cup D2 \cup \ldots \cup Dk = D$
2: For each partition $Di$, the KAP algorithm is performed to select the $n$ number of local exemplar set of this partition, denoted as $Ei$. Where $n$ is the number of clusters
3: Exemplars of all the partitions create a new data set, denoted as:
```

$E = E1 \cup E2 \cup \ldots .\cup Ek.$

IWC will be used on the data set E to select the global exemplars of the entire data set, denoted as $Eg1, Eg2, \ldots ., Egn$. Each exemplar Egi ($1 \leq i \leq n$) will be regarded as a centroid of cluster Ci, which is $Ci = \{ Egi \}$.

4: For each point di in data set D the similarities between di and each exemplar Egi, denoted as sim(di; Egi), are compared to find the exemplar point m with the maximal similarity.

$$m = \max_{j} sim(di; cj)$$

$$then\ Cm = Cm \cup \{di\}$$

5: Return the clustering result

$$D = C1 \cup C2 \cup \ldots .. \cup Cn$$

3.1.2 Key issues in the proposed algorithm

The Choosing partition size should address for the implementation of the algorithm.

The Choosing partition size: due to the limitations in time and memory consumption, the partition size should be decided by both the executing efficiency and the clustering result of the algorithm.

To get better results, the partitions of the data set should be a representative subset of the real data set. Usually, the bigger the partition size is, the better the representation of data will be, as well as the result of KAP. However, due to the memory and time consumption in KAP, the small partition size is preferred. Therefore, for proposed method 1 neither big nor small partition size would lead to better the results. The choice of the partition size should be based on; application's specific demands for the clustering result, the efficiency of the clustering algorithm, and the characteristics of the data set that will be clustered. Furthermore, well-divided partitions ensure that each partition is a good representation of the entire data set. For proposed method 2, the K-means algorithm can produce well-divided partitions that can be a good representation of the entire data set.

3.2 IDS Proposed Frameworks

In this section we present two hybrid IDS contributions; the first is using IWC-KAP Clustering with Naïve Bayes. The other is using IWC-KAP Clustering with Decision Tree.

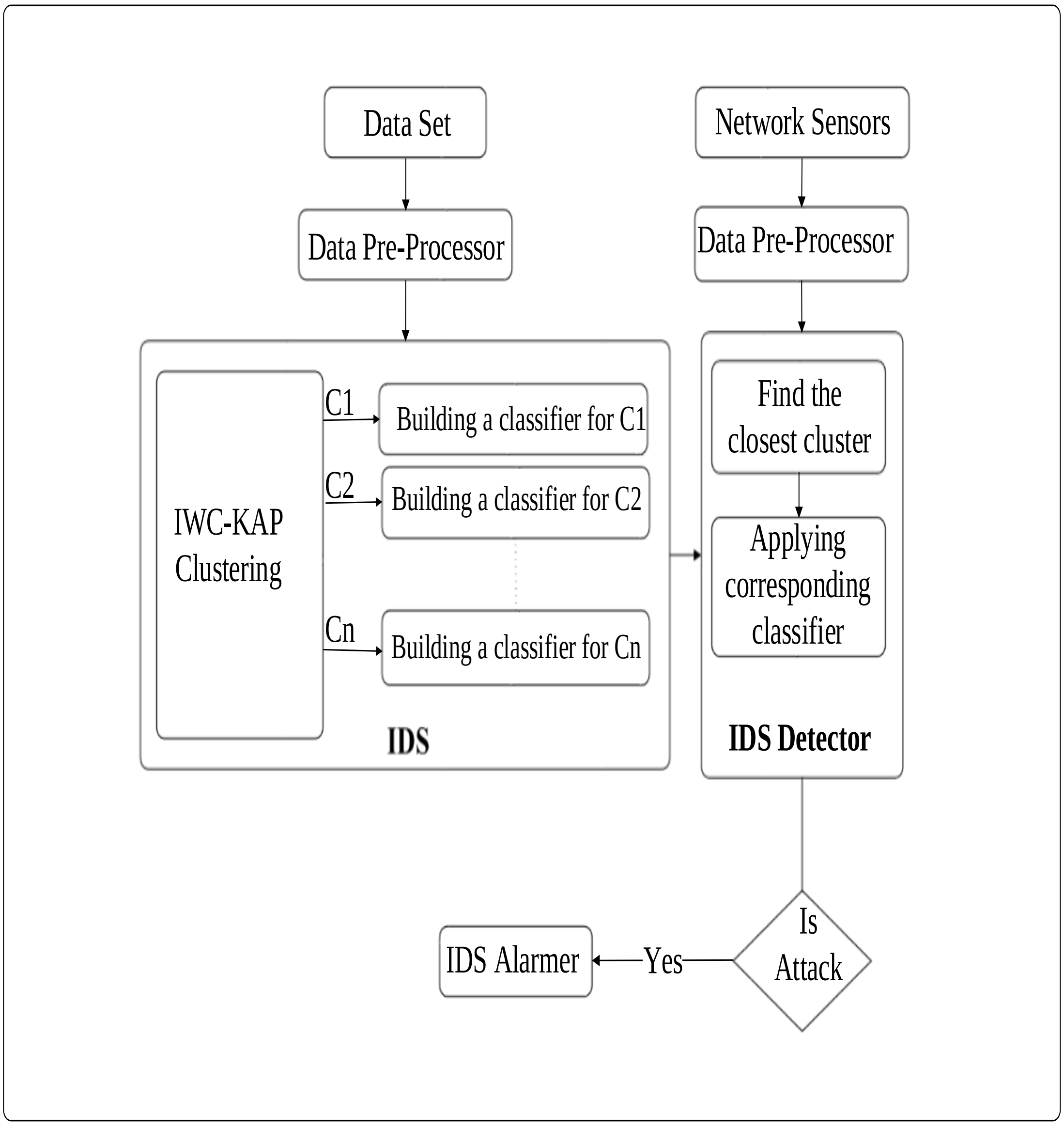

Figure 3.1 Proposed hybrid model IDS

Fig 3.1 shows the block diagram for the proposed framework. There are two phases in the framework: the clustering phase and the classification phase. After preprocessing the training data, the clustering algorithm cluster the data to k cluster. Then the corresponding classifier is built for each cluster. In the testing, network traffic is captured by the network sensors and fed to the IDS detector after being preprocessed. The IDS detector determines the relative cluster for the connection then apply the corresponding classifier for that cluster. The IDS detector raises an alarm if any connection classified as intrusion pattern. Figures 3.2 and 3.3 show the training and testing processes, respectively. The details for each model are shown in the rest of this section.

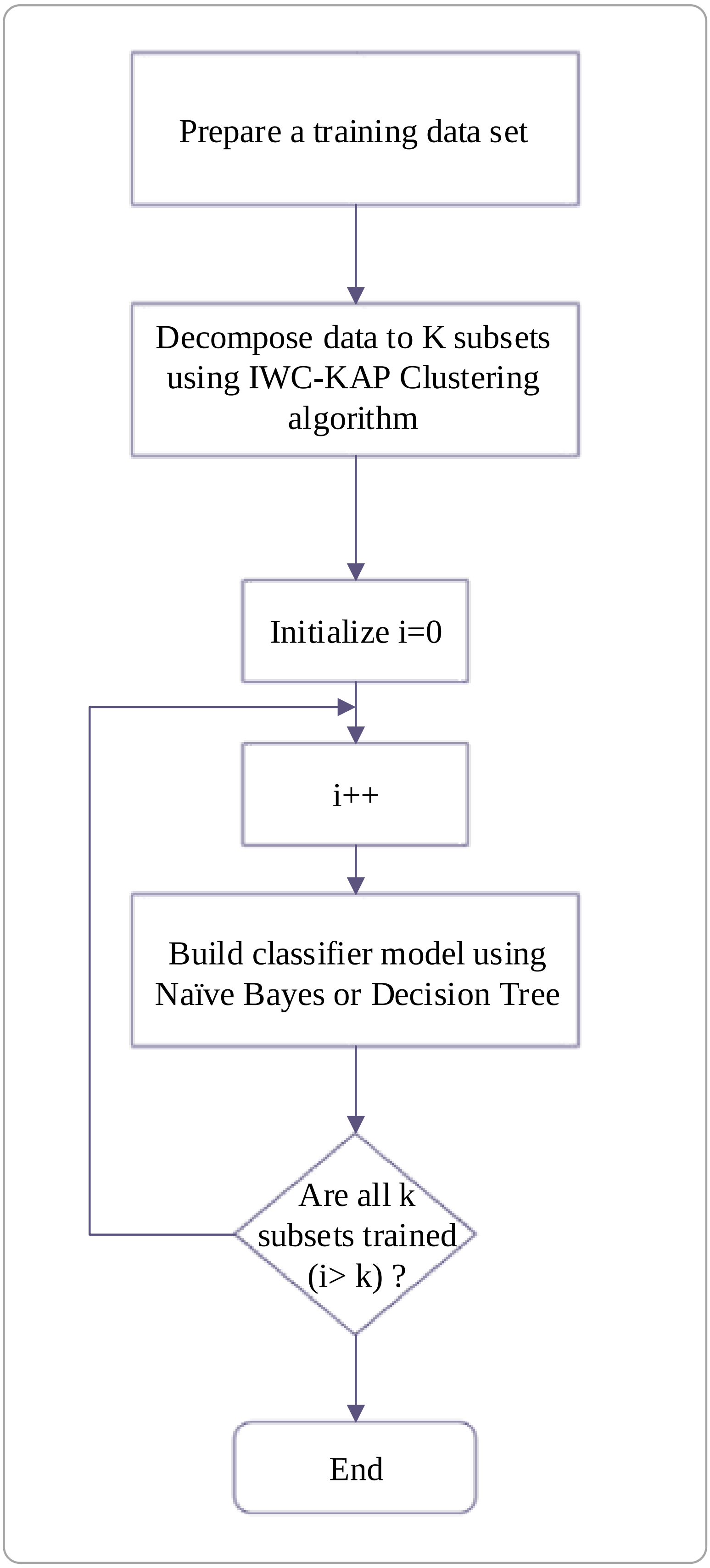
Prepare a training data set
Decompose data to K subsets using IWC-KAP Clustering algorithm
Initialize i=0
i++
Build classifier model using Naïve Bayes or Decision Tree
Are all k subsets trained (i> k) ?
End

Figure 3.2 Training process of the proposed IDS

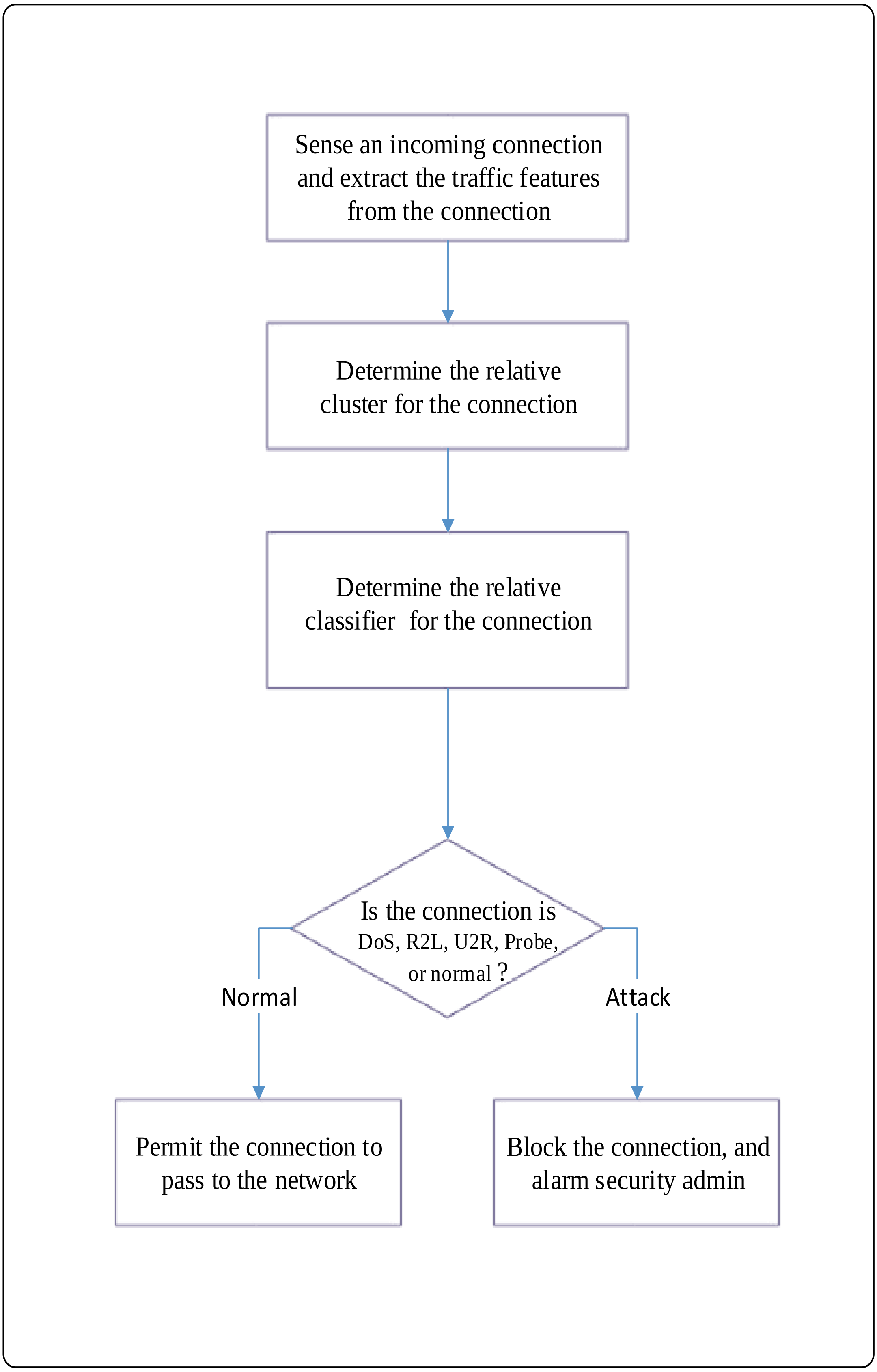
Sense an incoming connection and extract the traffic features from the connection
Determine the relative cluster for the connection
Determine the relative classifier for the connection
Is the connection is DoS, R2L, U2R, Probe, or normal ?
Normal
Attack
Permit the connection to pass to the network
Block the connection, and alarm security admin

Figure 3.3 Testing process of the proposed IDS

3..2.1 IDS using IWC-KAP and Naïve Bayes

While anomaly learning approaches can achieve high detection rate, false alarms rate is still high. This work is a combination of clustering and classification techniques that aim at maintaining the high detection rate and accuracy while reducing false alarm rate as much as possible.

IWC-KAP clustering is used as a pre-classification for the first stage in the proposed hybrid learning approach, to cluster similar data instances based on its behaviours. The basic idea of the IWC-KAP clustering is that the data set is divided into several subsets, each of which can be efficiently clustered by KAP. The resulting subsets exemplars are clustered through the IWC algorithm [39] to get a specific number of global exemplars, and then each data point has clustered to its exemplars.

Next, for the second stage clustered data will be classified using Naïve Bayes Classifier into attack classes. Thus, the data that have been misclassified during the first stage may be correctly classified in the second stage.

Network intrusion data is divided into four classes; Probe, DoS, U2R, and R2L, in addition to the normal data class.

The main goal for using IWC-KAP clustering is to split data into 5 classes. IWC-KAP clustering method identifies a set of 'exemplars' that represents the data set and partitions input data set into k-clusters. IWC-KAP attempt to find the exemplars that maximize the net similarity between each cluster. This work chooses the number of cluster k = 5 in order to cluster the data into five clusters (C_1, C_2, C_3, C_4, C_5).

A lot of clustering algorithms including IWC-KAP are unable to cluster correctly intrusion instances and normal instances because some of attacks behaviors are similar in intrusion instances and normal instances. To improve the shortcoming of the clustering algorithms, this work combined the IWC-KAP algorithms with Naïve Bayes Classifier. In data mining, Naïve Bayes has become one of the popular, efficient classification algorithms [76]. Naïve Bayes analyzes the relationship between dependent variable and the independent variable to derive a conditional probability for each relationship.

Using Bayes Theorem:

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)} \qquad (3.1)$$

Where X is the data record.

H is some hypothesis represents X data record that belongs to a specified class C.

P(H|X) is the probability that the hypothesis H holds, given the observed data record X.

P(H) is the prior probability that is independent of X.

P(X|H) the posterior probability of X conditioned on H.

For stage 2, Naïve Bayes Classifier has been used to classify all data from stage1 into more specific class. Five classes (C_1 = Normal, C_2 = DoS, C_3 = Probe, C_4 = R2L, and C_5 = U2R) are considered. Given X, we can predict C_1, C_2, C_3, C_4, and C_5. by Bayes rule in (3.2).

$$P(Ci|X) = \frac{\mathrm{P(X|Ci)\times P(Ci)}}{P(X)} \qquad (3.2)$$

Where C_i is the classes category and X is the data record. X may be divided into pieces of instances, say x1, x2, ..., xn that are related to the attributes X1, X2, ..., XN, respectively.

The probability obtained is shown in (3.3).

$$P(Ci|X) = \frac{\mathrm{P}(X1|\mathrm{Ci})\times\mathrm{P}(X2|\mathrm{Ci})\ldots\times\mathrm{P}(Xn|\mathrm{Ci})\times\mathrm{P(Ci)}}{P(X)} \qquad (3.3)$$

However, having strong dependencies among attributes may result in poor performance. So this work uses IWC-KAP clustering algorithms to improve the constraint of Naïve Bayes Classifier in terms of detection rate, accuracy, and false alarm.

The combination of IWC-KAP Clustering algorithms and Naïve Bayes Classifier shows an improvement compared to the Naïve Bayes classifier, as it increases detection rate, accuracy and reduces false alarms.

3.2.2 IDS using IWC-KAP and Decision Tree

The proposed method is combining of two phases, IWC-KAP clustering Phase, and the Decision Tree phase.

3.2.2.1 IWC-KAP clustering

As the previous method, IWC-KAP clustering used to split data into 5 classes

3.2.2.2 Decision Tree

In this phase, DT is built with the instances in each IWC-KAP cluster. The DT that trained on that cluster refines the decision boundaries by partitioning the instances with a set of if-then rules over the feature space. We used two decision tree algorithms the J48 and the J48-Graft algorithm.

During training, IWC-KAP clustering method is firstly applied to partition the training data into k disjoint clusters 1 2 3, ..., K. Then, DT is built for each cluster by using the J48 and or the J48-Graft techniques. The IWC-KAP method ensures that each training instance is associated with only one cluster.

In the testing, there are two phases selection and classification Phase. In the selection phase, compute the Euclidean distance for every testing instance and find the closest cluster. Then, chose The DT for the closest cluster. In classification phase, apply the test instance over the DT of the computed closest cluster and classify the test instance as normal or one of the four attack types. The algorithm for the proposed method as shown in algorithm 3.3 during the training and algorithm 3.4 during the testing.

Algorithm 3.3 The hybrid IWC-KAP and Decision Tree method during training

```
Clustering Phase
Input: Dataset
Output: Clusters
Procedure Clustering
Begin
Step 1: divide dataset to n part.
Step 2: for each part find the local exemplars.
Step 3: find global exemplars from local exemplars.
Step 3: Assign every instance $Z \in$ Dataset to the closest exemplar to make clusters {1,2,...,}
End

Classification Phase
Input: Clusters
Output: Decision tree
Procedure Classification
Begin
Step 1: Build the tree for each cluster
End
```

Algorithm 3.4 The hybrid IWC-KAP and Decision Tree method during testing

```
Selection Phase
Input: Test instances $Z_i$ and i = 1,2,3,......,N .
Output: Closest cluster to the test instance $Z_i$.

Procedure Selection

Begin
Step 1: For each test instance $Z_i$
    a. Compute the Euclidean distance $E(Z_i, R_j)$, j=1...k, and find the cluster closest to $Z_i$.
    b. Compute the J48 Decision Tree for the closest cluster.
End

Classification Phase
Input: Test instance $Z_i$.
Output: Classified test instance $Z_i$ as DoS, R2L, U2R, Probe, or normal.

Procedure Classification

Begin
Step 1: Apply the test instance $Z_i$ over the decision tree of the computed closest cluster.
Step 2: Classify the test instance $Z_i$ as DoS, R2L, U2R, Probe, or normal.
End
```

Chapter 4
Experimental Results

4.1 Experimental Results for IWC-KAP

Experiments are conducted on four clustering algorithms using a computer with 8G memory and 2.5GHz frequency. The algorithms are the proposed algorithm (i.e. IWC-KAP), traditional AP, KAP algorithm, and HAP algorithm. The experiments are set to illustrate whether the proposed algorithm is more suitable for the large-scale data set clustering problem than the other algorithms.

Before we use the IWC-KAP in the IDS proposed models we are test it on the traditional clustering data set and compare it with other clustering algorithms which depend on AP clustering to show the performance of IWC-KAP.

4.1.1 Data sets and generating methods

The data sets and their characteristics of each data set for the experiments are as shown in Table 4.1

Table 4.1 Characteristics of Clustering data sets

Data sets	Data size	Number of classes	Attribute dimension
IRIS	150	3	4
Wine	178	3	13
Yeast	1484	10	8
Ionosphere	151	2	34
Heart	302	5	13
S-Data1	3500	7	2
S-Data2	1800	6	2
S-Data3	1400	7	2

4.1.1.1 Artificial data set

Three artificial two-dimensional data set S-Data1, S-Data2, and S-Data3, are generated using the random function in Matlab. As shown in Table 4.1, S-Data1 contains 3500 samples, S-Data2 contains 1800 samples and S-Data3 contains 1400 samples. The data points of the artificial data sets are described in two-dimensional attributes to simplify the computation without loss of generality.

4.1.1.2 Real data set

As described in Table 4.1, five real data sets are used

Iris: 150 samples with 4 dimensions and 3 clusters.

Yeast: 1484 samples with 8 dimensions and 10 clusters.

Wine: 178 samples with 13 dimensions and 3 clusters.

Ionosphere: 151 samples with 34 dimensions and 2 clusters.

Heart: 302 samples with 13 dimensions and 5 clusters.

These data sets are used in most clustering algorithm experiments, and can be obtained from the UCI machine learning knowledge base website [77].

This work used the previous real data set to compare the result with [78] who used the same data set.

Euclidean distance method is used to find the similarity between data points pi and pj in the data set. Distance is described as in the following expression.

$$s(pi, pj) = -|| pi - pj||^2 \qquad (4.1)$$

The partition size of the data used to test algorithm 3.1 is selected as 0.25 times of the data set size that will cluster. However, the partition size for algorithm 3.2 depends on the size of initial clusters result from K-means algorithm. For the partition step in algorithm 3.1, the code was run 50 times. The results were recorded, and the average value was calculated.

4.1.2 Evaluation Methods

The Normalized Mutual Information (NMI) index [79] is used to evaluate the results of the four algorithms; AP, KAP, HAP and the proposed algorithm. NMI is used to measure the similarity between the result of the clustering algorithm and the standard division of the data set. The calculation of the NMI index can be described as follows:

For any partition of the data set, denoted as P^a, the information entropy (which is the expected value (average) of the information) of this partition is:

$$H(P^a)=-\sum_{i=1}^{ka}\frac{n_i^a}{n}\log(\frac{n_i^a}{n}) \qquad (4.2)$$

Where n is the data size, ka is the number of clusters in the partition, $n^a{}_i$ is the number of points in the i-th cluster in the partition.

The mutual information (which is a measure of the variables' mutual dependence) for two partitions of the same data set *Pa* and *Pb*, is calculated by the following formula:

$$I(p^a,p^b)= \sum_{i=1}^{ka}\sum_{j=1}^{kb}\frac{n_{ij}^{ab}}{n}\log(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_i^a}{n}\times\frac{n_j^b}{n}}) \qquad (4.3)$$

Where n_{ij}^{ab} is the number of the points both in the *i*-th cluster of P^a and in the *j*-th cluster of P^b.

The lack of information between two vectors is defined as:

$$I(p^a \mid p^b)=-\sum_{i=1}^{ka}\sum_{j=1}^{kb}\frac{n_{ij}^{ab}}{n}\log(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_j^b}{n}}) \qquad (4.4)$$

From the communication theory point of view, the above-defined quantities can be interpreted as follows. Suppose we need to transmit all the cluster labels in P^a on a communication channel, then H(P^a) can be interpreted as the average amount of information, for example, in bits, needed to encode the cluster label of each data point according to P^a. Now suppose that P^b is made available to the receiver, and then H(P^a | P^b) denotes the average number of bits needed to transmit each label in p^a if P^b is already known. We are interested to see how much H(P^a | P^b) is smaller than H(P^a), that is, how much the knowledge of P^b helps us to reduce the number of bits needed to encode P^a. This can be quantified in terms of the mutual information H(P^a)−H(P^a | P^b) = I(P^a, P^b). The knowledge of P^b thus helps us to reduce the number of bits needed to encode each cluster label in P^a by an amount of I(P^a , P^b) bits. In the reverse direction, we also have I(P^a, P^b) = H(P^b)−H(P^b P^a). Clearly, the higher the MI, the more useful the information in P^b helps us to predict the cluster labels in P^a and vice-versa.

The similarity of two partitions *Pa* and *Pb* for the same data set is calculated using the NMI index as follows.

$$NMI(p^a,p^b) =\frac{2\times I(pa,pb)}{H(Pa)+H(Pa)} \qquad (4.5)$$

The NMI measures the information that P^a and P^b share: it tells us how much each one of these clusters reduces our uncertainty about the other. The value of the NMI index for two partitions of any data set is [0..1]. The bigger the NMI index is, the more similarity the two partition are.

4.1.3 Results and Analysis

Table 4.2, Fig 4.1 and Fig 4.2, show the clustering time of the real and the artificial data sets, and compare algorithm 3.1 and algorithm 3.2 with the AP, KAP and HAP algorithms.

Table 4.3, Fig 4.3 and Fig 4.4, show the NMI index of the real and the artificial data sets, and compare algorithm 3.1 and algorithm 3.2 with the AP, KAP and HAP algorithms.

Table 4.2 Clustering time (Sec) IWC-KAP Vs. AP, KAP, and HAP

	AP	HAP	KAP	Algorithm 3.1 (Best)	Algorithm 3.1 (Worst)	Algorithm 3.1 (Avg)	Algorithm 3.2
Yeast	444.71	360.48	334.81	57.16	90.23	73.70	139.09
Wine	7.33	17.35	6.58	5.05	7.09	6.07	6.20
Iris	13.36	8.27	4.06	4.66	5.37	5.01	3.68
Heart	24.81	19.27	17.66	2.36	4.63	3.49	1.11
Ionosphere	12.34	10.99	2.78	0.39	0.72	0.56	0.29
S-Data1	233.65	58.41	165.96	41.49	54.47	47.98	44.75
S-Data2	82.83	18.83	65.97	8.13	10.68	9.41	8.54
S-Data3	17.01	8.04	15.22	5.53	6.69	6.11	6.40

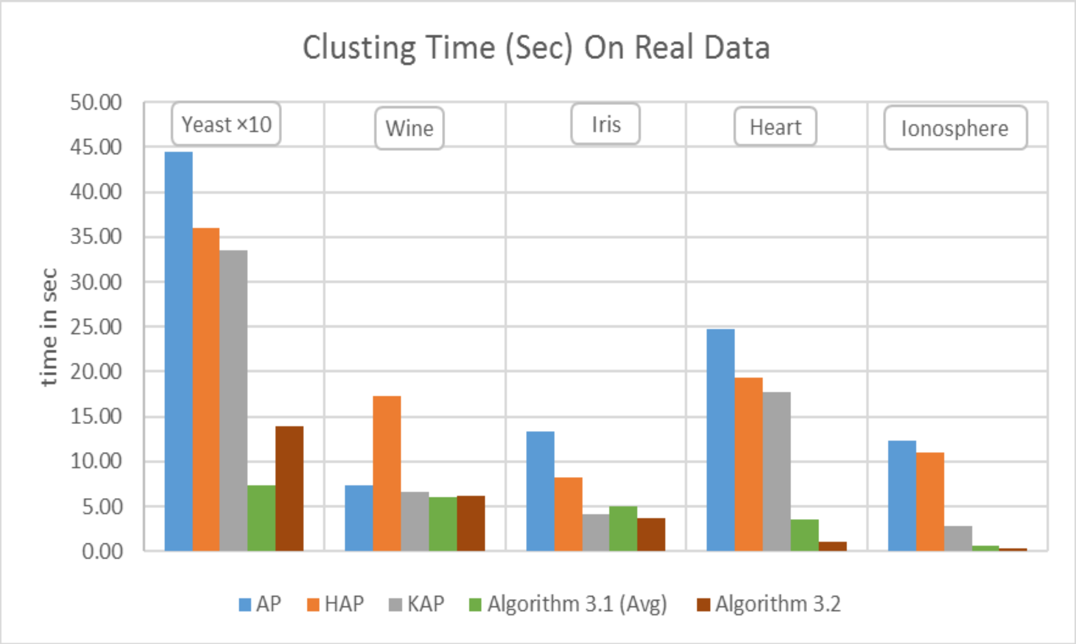

Figure 4.1 Clustering time IWC-KAP Vs. AP, KAP, and HAP in real data set.

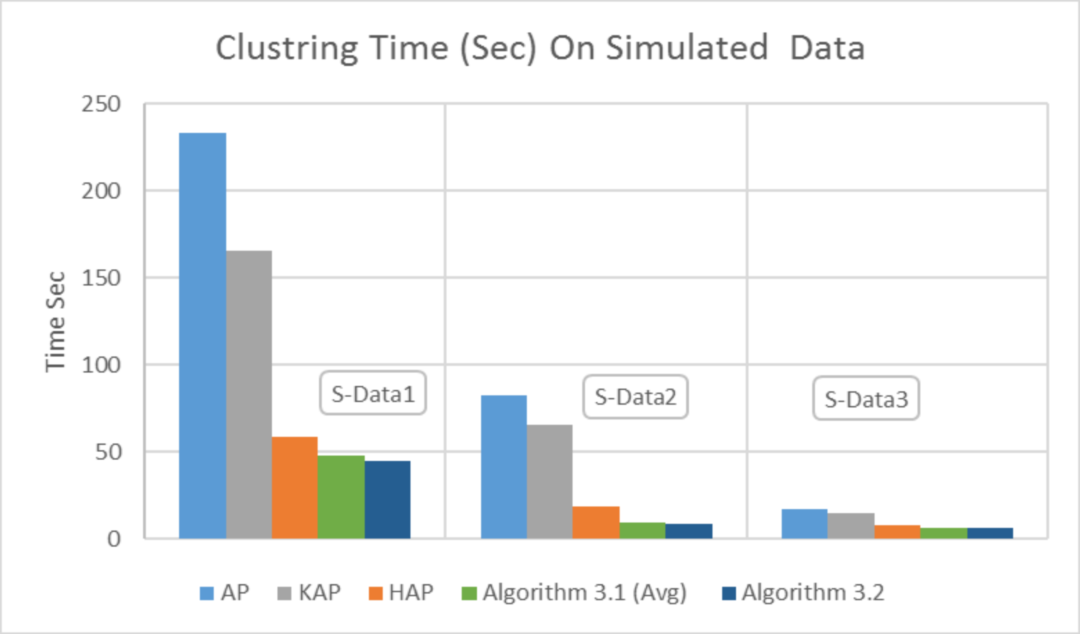

Figure 4.2 Clustering time IWC-KAP Vs. AP, KAP, and HAP in Artificial data set.

Table 4.3 NMI index IWC-KAP Vs. AP, KAP, and HAP

	AP	KAP	HAP	Algorithm 3.1 (Best)	Algorithm 3.1 (Worst)	Algorithm 3.1 (Avg)	Algorithm 3.2
Iris	0.50	0.85	0.75	0.82	0.66	0.74	0.71
Wine	0.31	0.38	0.41	0.46	0.39	0.43	0.42
Yeast	0.28	0.39	0.18	0.28	0.20	0.24	0.24
Ionosphere	0.13	0.13	0.09	0.13	0.03	0.08	0.08
S-Data1	0.90	0.91	0.90	0.91	0.90	0.91	0.91
S-Data2	0.91	0.93	0.93	0.95	0.90	0.93	0.95
S-Data3	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Through Table 4.2, Table 4.3, Fig 4.1 till Fig 4.4, it can be seen that algorithm 3.1 and algorithm 3.2 achieve the lowest time consumption compared to AP and KAP algorithms for almost all data sets. Furthermore, algorithm 3.1 and algorithm 3.2 gave better results than AP with the NMI index and gave an almost result as close as possible to KAP algorithm. However, time consumption increases rapidly with the growth of the data size in the AP and KAP algorithm, which makes those algorithms not suitable for solving the clustering problem of large-scale data. In algorithm 3.1 and algorithm 3.2, the clustering results are significantly better than AP and KAP on all the data sets except the Ionosphere and Yeast data. Also, the time consumption is much lower than AP and KAP on all data. The great time consumption of AP and KAP is due to the computation of matrixes of similarities between pairs of data points in the entire data set while the input to algorithm 3.1 and algorithm 3.2 is the matrix of similarities between pairs of data points only in each partition, and the input to the selection of global exemplars is the matrix of its data record only. algorithm 3.1 and algorithm 3.2 have advantages over others in term of memory consumption comparing with AP and KAP. It is also more efficient and accurate more than the others.

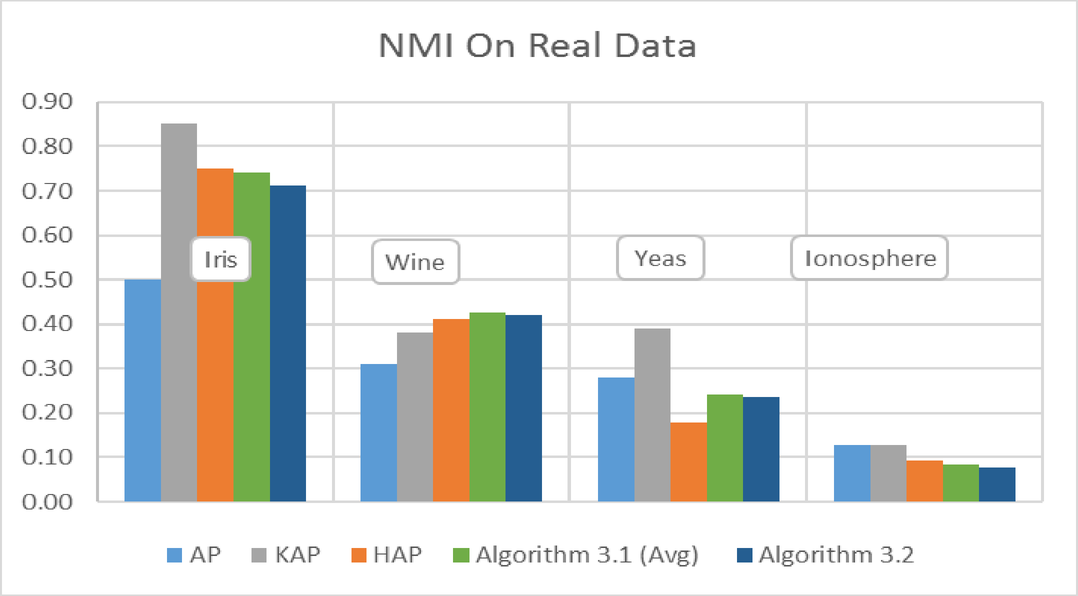

Figure 4.3 NMI of IWC-KAP Vs. AP, KAP, and HAP in real data set.

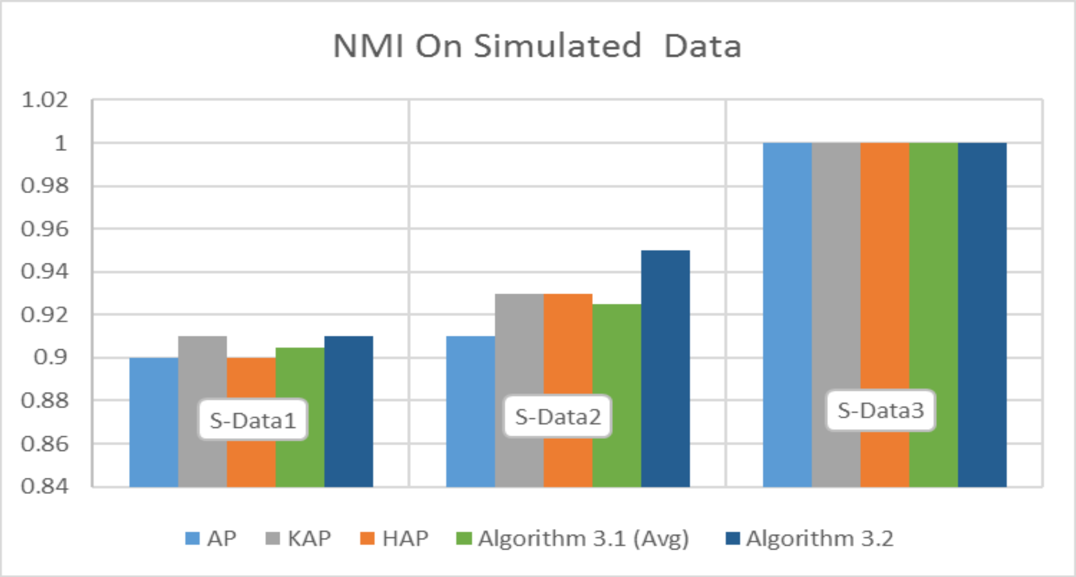

Figure 4.4 NMI of IWC-KAP Vs. AP, KAP, and HAP in Artificial data set.

We also compare algorithm 3.1 and algorithm 3.2 with the HAP algorithm, which is the first algorithm that addressed the AP problem in the large-scale data set. As it can be seen in Table 4.2, Fig 4.1 and Fig 4.2 the time consumption of the proposed algorithm is not only lower than AP and KAP algorithms but also lower than HAP algorithm. However, as seen in Table 4.3, Fig 4.3 and Fig 4.4, the clustering results by the NMI index is almost close to HAP algorithm. However, the time consumption increases rapidly with the growth of the data size in the HAP, which makes the proposed algorithms more efficient as it consumes time less than HAP when the size of data grow. The increase in time consumption of HAP results is due to the computation of the matrices of similarities between pairs of data points in the entire data set when using the AP algorithms again to get the global exemplars from the local exemplars. However, the proposed algorithms use modified K-means algorithms to find global exemplars that decrease the time because the K-means is faster than AP algorithms. The proposed algorithms have advantages over HAP when it comes to less memory consumption. It is also more efficient and accurate.

4.2 IDS Experimental Results

This section displays the results of the two hybrid IDS that are mentioned in chapter 3, we will show the parameter optimization for each model and discuss the obtained results.

4.2.1 Dataset

Currently, there are only a few public datasets for network-based IDSs like KDD Cup ’99[1], the majority of the experiments in the intrusion detection domain performed on these datasets [80].

Datasets that are used for intrusion detection are categorized into three categories DARPA, KDD Cup, and some real world datasets. The KDD Cup dataset is widely used by the researchers to test the effectiveness of the developed method for intrusion detection with 42%, 20 % of the studied papers used DARPA dataset to check the effectiveness of the methods for intrusion detection, and the rest of the studied papers used other real world data sets [17].

KDD Cup dataset contains training data that include seven weeks of network traffic in the form of TCP dump data consisting of approximately 5 million connection records, each of which is approximately 100 bytes. The test data included two weeks of traffic, with approximately 2 million connection records [80].

[1] Dataset KDD Cup is available on web site http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; different version is available for supervised and unsupervised learning, also reduced version 10% KDD Cup is available which is used in our study.

Table 4.4 Mapping attack types to the attack classes on KDD Cup '99 dataset [36]

Class	Attacks in the training data	Additional Attacks in the testing data
DOS	Back, land, Neptune, pod, smurf, teardrop	apache2, mailbomb, processtable, udpstorm
Probe	ipsweep, portsweep, satan, nmap	mscan, saint
U2R	buffer_overflow, loadmodule, rootkit, perl	httptunnel, ps, worm, xterm
R2L	ftp_write, guess_passwd, imap, multihop, warezmaster, warezclient, spy, phf	named, sendmail, snmpgetattak, snmpguess, sqlattack, xlock, xsnoop

This work used the 10% KDD Cup'99 benchmark dataset KDD [80] for evaluation and comparison between the proposed approaches and the previous approaches. The entire 10% KDD data set contains an approximately 500,000 instances with 41 features. The training dataset contains 24 types of attacks while the testing data contains more than 14 types of additional attack. All of which are mapped to four basic attack classes as shown in Table 4.4 [15].

To show the ability to detect different kinds of intrusions, the training and testing data covers four major categories of attacks as follows:

- Denial of Service attacks (DoS)
- Remote to Local (User) Attacks (R2L)
- User to Root Attacks (U2R)
- Probing

Table 4.5 Training dataset sample distribution

Class	No. of Samples	Sample Percentage (%)
Normal	97277	19.69
Probe	4107	0.83
DoS	391458	79.24
U2R	52	0.01
R2L	1126	0.23
Total	494020	100

Table 4.6 Testing dataset sample distribution

Class	No. of Samples	Sample Percentage (%)
Normal	60593	19.4
Probe	4166	1.33
DoS	231455	74.4
U2R	88	0.028
R2L	14727	4.73
Total	311029	100

Tables 4.5 shows “10% of KDD Cup ‘99" distribution records as training dataset by class type. While Table 4.6 shows the testing dataset records.

4.2.2 Software and Tools

In our experiments, we use Weka tool[2] [81] (Waikato Environment for Knowledge Analysis) and Matlab to evaluate our proposed methods. Weka is an open source software written in java, it has a collection of machine learning algorithms for data mining tasks and is widely used for teaching and researching.

4.2.3 Evaluation Measurement:

Regarding to the previous researches in IDS area, the performance of IDS is measured and evaluated by calculated confusion matrix, Table 4.7 shows the components of the confusion matrix.

Table 4.7 Confusion Matrix

Actual	Predicted Normal	Predicted Attack
Normal	True negative (TN)	False positive (FP)
Attacks	False negative (FN)	True positive (TP)

- True positive (TP) when attack data is detected as an attack.
- True negative (TN) when normal data is detected as normal.
- False positive (FP) when normal data is detected as an attack.
- False negative (FN) when attack data is detected as normal.

[2] Weka software and source code is available to download from http://www.cs.waikato.ac.nz/~ml/weka/ website for both x32 and x64 bit computers

A practical IDS is characterized by its high detection rate and low false alarm rate. In general, the performance of the IDS is evaluated in terms of detection rate, accuracy and false alarm rate as in the following formula:

- Detection Rate = (TP) / (TP+FP)
- False Alarm = (FP) / (FP+TN)
- Accuracy = (TP+TN) / (TP+TN+FP+FN)

4.2.4 Data Preparation

KDD Cup is multiclass datasets; the original datasets files downloaded from website labels the data by attack type (i.e. normal, nepton, snmpattack, and so on). We divide connections to five classes according to Table 4.4, in addition to the normal class.

4.2.5 Results of IDS using NB

This section shows and discusses the results obtained using IWC-KAP with NB and NB Multinomial.

Table 4.8 IDS using NB classification results

Methods	NB	KM+NB	IWC-KAP+NB	IWC-KAP+NB Multinomial
Group 1	81.00%	99.50%	99.21%	99.51%
Group 2	95.60%	98.30%	95.62%	97.82%
Group 3	82.50%	99.60%	99.57%	99.74%
Group 4	80.00%	80.00%	92.26%	92.68%
Group 5	90.30%	83.20%	92.88%	93.52%
Accuracy	83.19%	99.00%	99.13%	99.38%
Detection Rate	94.70%	98.80%	99.51%	99.68%
False Alarm	19.00%	2.20%	1.29%	0.79%
False Alarm	19.00%	2.20%	1.29%	0.79%

Table 4.8 and Fig 4.5 show the results across all category classes obtained from Naïve Bayes (NB), hybrid learning approach K-Means with Naïve Bayes (KM+NB), hybrid

learning approach IWC-KAP with Naïve Bayes (IWC-KAP+NB) and hybrid learning approach IWC-KAP with Naïve Bayes Multinomial (IWC-KAP+NB Multinomial). IWC-KAP+NB and IWC-KAP+NB Multinomial have been deployed as in a single run. IWC-KAP+NB and IWC-KAP+NB Multinomial performed better than the single classifier NB and KM+NB in detecting Normal, U2R, R2L, and DoS instances. Meanwhile, IWC-KAP+NB and IWC-KAP+NB yield to good result in Probe instances.

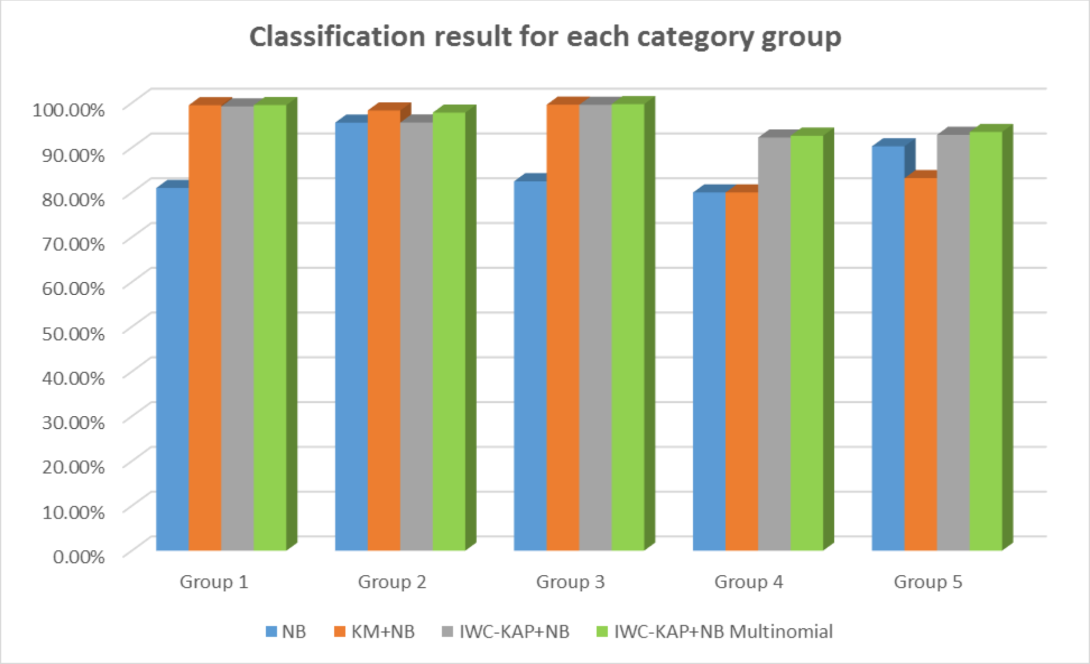

Figure 4.5 IDS using NB classification result for each category class

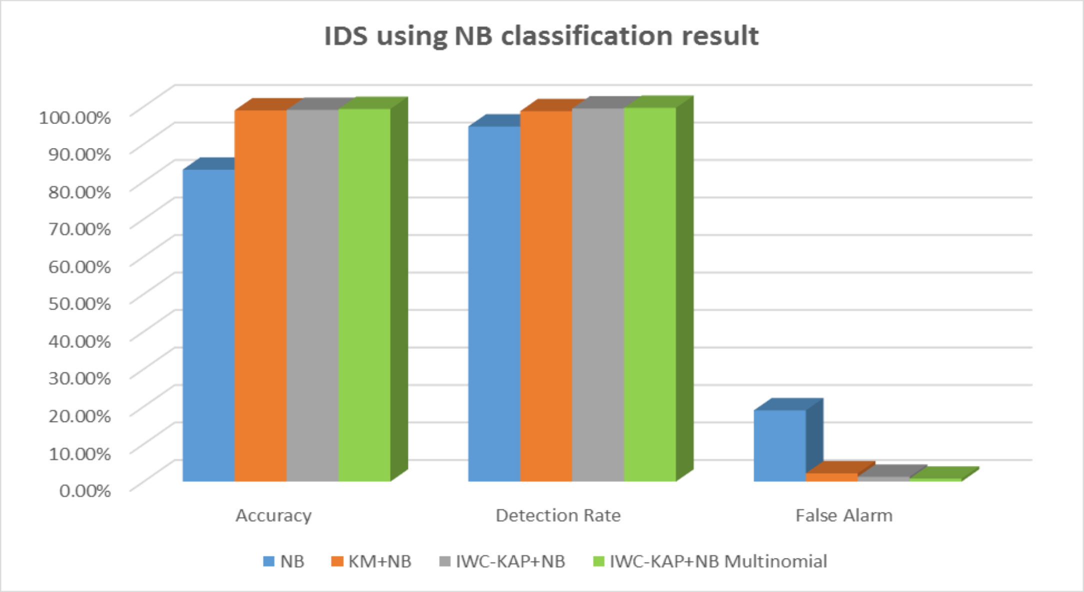

Figure 4.6 IDS using NB classification result

Table 4.8 and Fig 4.6 show the results of applying Naïve Bayes (NB), hybrid learning approach K-Means with Naïve Bayes (KM+NB), hybrid learning approach IWC-KAP with Naïve Bayes (IWC-KAP+NB) and hybrid learning approach IWC-KAP with Naïve Bayes Multinomial (IWC-KAP+NB Multinomial) on testing sets. The results are given in terms of detection rate, accuracy, and false alarm. It can be seen that while NB produced a slightly higher accuracy and detection rate, false alarm rates is still high. Meanwhile, the hybrid approaches IWC-KAP+NB and IWC-KAP+NB Multinomial recorded high accuracy and detection rate with low false alarm rate. This indicates that the proposed hybrid approaches perform better than KM+NB. The IWC-KAP clustering technique that was used in pre-classification component – for clustering similar data into a respective cluster helped IWC-KAP+NB and IWC-KAP+NB Multinomial to produce better results compared to NB and KM+NB. Additionally, the hybrid approach allows misclassified data during the first stage to be re-classified, hence improving the accuracy and detection rate with acceptable false alarms.

4.2.6 Results of IDS using DT Result and Discussion

Table 4.9 IDS using DT classification results

Methods	J48	J48-Graft	KM+J48	KM+1R	MAS-IDS	IWC-KAP+J48	IWC-KAP+J48-Graft
Group 1	89.23%	92.23%	90.56%	99.32%	91.63%	99.50%	99.45%
Group 2	87.23%	94.25%	93.62%	99.21%	93.32%	99.18%	99.35%
Group 3	84.62%	87.36%	87.65%	98.36%	97.32%	99.67%	99.99%
Group 4	82.23%	90.23%	92.07%	91.96%	90.32%	92.07%	93.09%
Group 5	83.56%	89.36%	91.36%	92.32%	91.65%	93.84%	93.62%
Accuracy	85.13%	90.17%	90.21%	99.26%	91.13%	99.35%	99.57%
Detection Rate	72.98%	81.32%	83.94%	99.33%	85.26%	99.50%	99.45%
False Alarm	2.73%	0.81%	3.53%	2.73%	2.99%	0.42%	0.43%

Table 4.9 and Fig 4.7 show the results across all category classes obtained from hybrid learning approach IWC-KAP with the J48 Classifier (IWC-KAP+J48) and hybrid learning approach IWC-KAP with J48Graft Classifier (IWC-KAP+J48Graft).

IWC-KAP+J48 and IWC-KAP+J48Graft have been deployed as in a single running. IWC-KAP+J48 and IWC-KAP+J48Graft performed a good result in all classes.

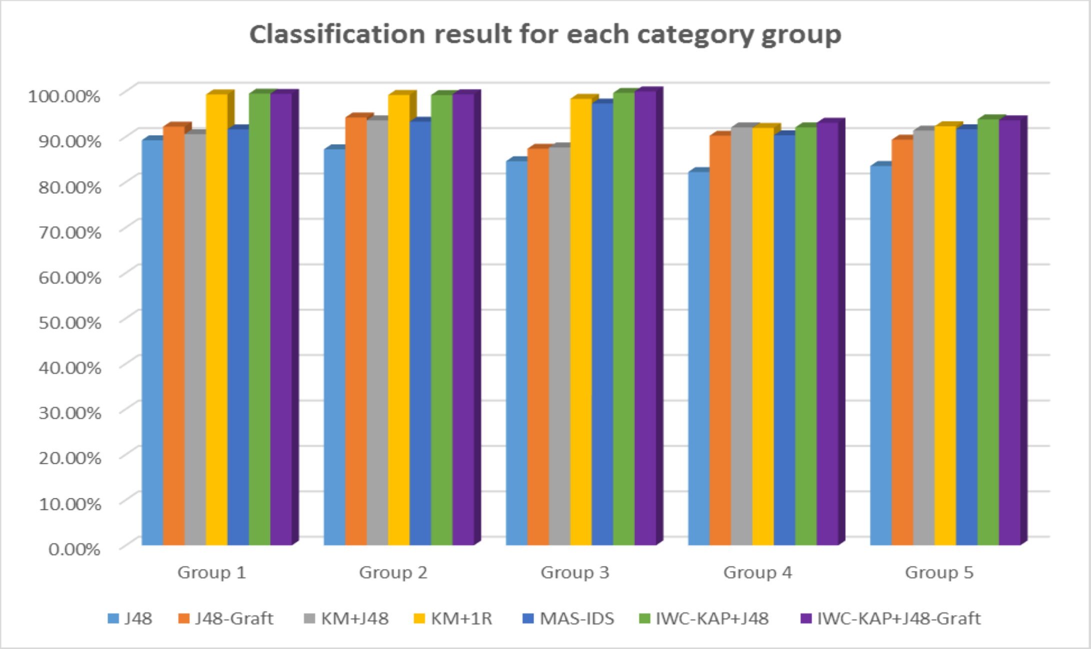

Figure 4.7 IDS using DT classification result for each category class

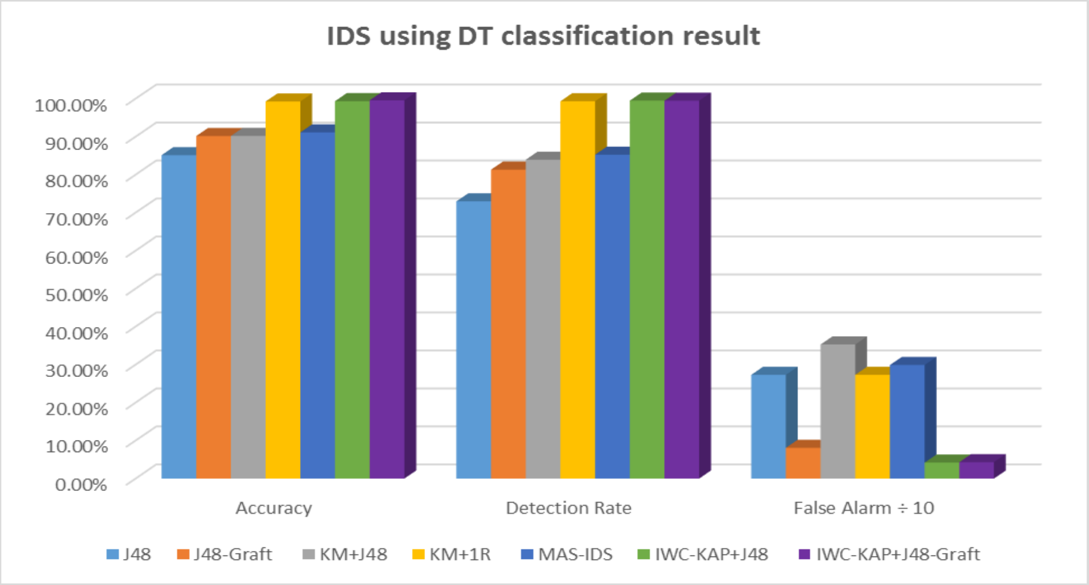

Figure 4.8 IDS using DT classification result

Table 4.9 and Fig 4.8 show the results of applying J48, J48graft, hybrid learning approach K-means with J48 (KM+J48), hybrid learning approach modified K-means with J48 (MAS-IDS), hybrid learning approach modified K-means with One-R Classifier (KM-1R), hybrid learning approach IWC-KAP with J48 Classifier (IWC-

KAP+J48) and hybrid learning approach IWC-KAP with J48-Graft Classifier (IWC-KAP+J48-Graft) on testing sets. The results are given in terms of detection rate, accuracy, and false alarm. It can be seen that while J48 and J48graft produced a slightly higher accuracy and detection rate, false alarm rates are still high. Meanwhile, the hybrid approaches IWC-KAP+J48 and IWC-KAP+J48-Graft recorded high accuracy and detection rate with low false alarm rate. This indicates that the proposed hybrid approaches perform better than KM+J48, MAS-IDS, and KM-1R. The IWC-KAP clustering technique that was used in pre-classification component – for clustering similar data into a respective cluster, IWC-KAP+ J48 and IWC-KAP+ J48-Graft to produce better results compared to other approaches. Additionally, the hybrid approach allows misclassified data during the first stage to be re-classified, hence improving the accuracy and detection rate with acceptable false alarms.

Chapter 5

Conclusion and Future Work

In this research, large-scale data clustering algorithm and two hybrid network intrusion detection systems are proposed. Firstly, two large-scale data clustering methods depend on KAP and IWC algorithms were proposed. The proposed methods achieve efficient, accurate, and time-saving clustering for large-scale data sets. Data points were clustered by splitting the data into small groups, then KAP algorithm was implemented on each subset of the data. Then, IWC was applied to find the global exemplars for original finally well-suite clusters. These clusters were obtained by setting the points to its similar exemplars due to similarity function. The algorithms were tested using real and simulated data sets. The results showed that the proposed algorithms were more effective and efficient in term of clustering time consumption and memory space consumption than AP, KAP, and HAP. This was due to the proposed novel techniques. Secondly, a hybrid learning approach combines IWC-KAP clustering algorithm and Naïve Bayes classifiers (IWC-KAP+NB), and IWC-KAP clustering algorithm and Naïve Bayes Multinomial classifiers (IWC-KAP+NB Multinomial) were proposed. The proposed approaches were compared and evaluated using KDD Cup ’99 benchmark dataset. The proposed approaches separated instances between potential attacks and normal instances during the preliminary stage into different groups. Later, the groups were classified into more specific categories, Normal, DoS, Probe, R2L, and U2R.The hybrid learning approaches significantly reduced false alarm rates with an average below than 0.8%, while keeping detection and accuracy rate higher than 99%. The approaches can classify all data correctly in a higher percentage. Finally, a hybrid learning approach based on IWC-KAP clustering algorithm and decision trees classifiers was proposed. The proposed approach was compared and evaluated using KDD Cup ’99 benchmark dataset. The proposed approach separates instances into different cluster depend on its behavior during preliminary stage. Later, the groups were classified into more specific categories namely Normal, DoS, Probe, R2L, and U2R.The hybrid learning approaches significantly reduced false alarm rates with an average below than 0.5%, while keeping detection and accuracy rate higher than 99%.

In the future, other modified versions of AP will be used in clustering subsets instead of KAP. Furthermore, other partitioning algorithms are going to be used instead of IWC to find out whether better results can be achieved.

References

[1] M. Zulkernine and A. Haque, "Random-Forests-Based Network Intrusion Detection Systems," *IEEE Trans. Syst. Man. Cybern.*, Vol. 38, No. 5, pp. 649–659, 2008.

[2] S. X. Wu and W. Banzhaf, "The use of computational intelligence in intrusion detection systems: A review," *Applied Soft Computing Journal*, Vol. 10, No. 1, pp. 1–35, 2010.

[3] H. J. Liao, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, Vol. 36, No. 1, pp. 16-24, 2013.

[4] U. Aickelin and S. Cayzer, "The Danger Theory and Its Application to Artificial Immune Systems," *in 1st International Conference on Artificial Immune Systems ICARIS*, pp. 141–148, Canterbury, UK, 2002.

[5] H. Liu and H. Motoda, "Computational Methods of Feature Selection," *IEEE Intelligent Informatics Bulletin*, Vol. 9, No.1, pp. 39-40, 2008

[6] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, Vol. 28, No. 1, pp. 18–28, 2009.

[7] M. Mohammadi, Z. Muda, W. Yassin and N.I. Udzir, "KM-NEU: An Efficient Hybrid Approach for Intrusion Detection System," *Research Journal of Information Technology*, Vol. 6, No.1, pp.46–57, 2014.

[8] Z. Muda, W. Yassin, M.N. Sulaiman, N.I. Udzir, "K-Means Clustering and Naive Bayes Classification for Intrusion Detection," *Information Technology in Asia (CITA 11), 2011 7th International Conference on*, pp. 1-6, Kuching, Sarawak, 2011.

[9] J.S. Jang and C-T. Sun, "A Neuro-Fuzzy Classifier and its Applications," *Second IEEE International Conference on Fuzzy Systems*, Vol.1, pp 94-98, San Francisco, CA, 1993.

[10] Y. Liu, K. Chen, X. Liao W. Zhang, "A genetic clustering method for intrusion detection," *Pattern Recognition*, Vol. 37, No. 5, pp.927– 942, 2004.

[11] K. Latifur, A. Mamoun and T. Bhavani, "A New Intrusion Detection System Using Support Vector Machines And Hierarchical Clustering," *The VLDB Journal -The International Journal on Very Large Data Bases*, Vol. 16, No. 4, pp.507-521, 2007.

[12] S. V Sabnani, "Computer Security : A Machine Learning Approach", Technical Report, University of London, 2008.

[13] R. L. Krutz and J. Conley, *Wiley Pathways Network Security Fundamentals*. John Wiley & Sons, 2007, p. 524.

[14] M. Bishop, *Introduction to Computer Security*, Prentice Hall PTR, 2004, p. 784.

[15] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP '99 data set," *Computational Intelligence for Security and Defense Applications, IEEE Symposium on*, pp. 1-6, Ottawa, ON, 2009.

[16] Microsoft, "Microsoft's STRIDE threat model.", https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx, 2015.

[17] D. D. Clark and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *IEEE Symposium on Security and Privacy*, pp. 184–194, 1987.

[18] T. Mamatha and M. Z. Hussain, "Network Security Solutions and Vulnerabilities in E-Government," *International Journal of Computing and Business Research (IJCBR)*, ISSN (Online): 2229-6166, Vol 3, No. 3, 2012.

[19] S. Singh and S. Silakari, "A Survey of Cyber Attack Detection Systems," *IJCSNS International Journal of Computer Science and Network Security*, VOL.9, No.5, 2009.

[20] V. P. Sharma, R. Tiwari and R. K. Varun, "Dendritic Cell Algorithm, Dempster Belief Theory, Network Intrusion Detection System," *IOSR Journal of Computer Engineering*, Vol 16, No. 2, PP 99-103, 2014.

[21] H. Sarbazi-Azad, B. Parhami, S. G. Miremadi and S. Hessabi, *Advances in Computer Science and Engineering*, 2009.

[22] M. Maloof, *Machine learning and data mining for computer security*. Springer, 2006.

[23] C. Azad and V. K. Jha, "Data Mining in Intrusion Detection: A Comparative Study of Methods, Types and Data Sets," *Int. J. Inf. Technol. Comput. Sci*, Vol. 5, No. 8, pp. 75–90, 2013.

[24] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion detection systems," *Comput. Networks*, Vol. 31, No. 9, pp. 805–822, 1999.

[25] W. Stallings, *Network Security Essentials: Applications and Standards (3rd Edition)*, Jul. 2006.

[26] Wenke Lee, J. Salvatore Stolfo, and W. Kui Mok, "A Data Mining Framework for Adaptive Intrusion Detection, Proceedings," *lEEE Symposium on Security and Privacy*, p. 120-132, 1999.

[27] D. K. Kang, D. Fuller, and V. Honavar, "Learning classifiers for misuse and anomaly detection using a bag of system calls representation," *Information Assurance Workshop, IAW'05. Proceedings from the Sixth Annual IEEE SMC*, p. 118 – 125, 2005.

[28] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications,* Vol. 41, No. 4, pp. 1690-1700, 2014.

[29] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Systems with Applications: An International Journal,* Vol. 29, No.4, pp. 713–722, 2005.

[30] M. Solanki and V. Dhamdhere, "Intrusion Detection System by using K-Means clustering, C 4.5, FNN, SVM classifier," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* Vol. 3, No. 6, pp19-23, 2014.

[31] S. Borah, and A. Chakraborty, "Towards the Development of an Efficient Intrusion Detection System," *International Journal of Computer Applications,* Vol. 90, No.8, pp. 15-20, 2014.

[32] Y. Wang, *Statistical Opportunities, Roles, and Challenges in Network Security*, Yale University, USA, 2008.

[33] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," *Proc. Twenty-eighth Australas*. vol. 38, No.1, pp. 333-342, 2005.

[34] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," *Ain Shams Engineering Journal*, Vol. 4, No.4, pp. 753-762, 2013.

[35] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," in: *Applications of data mining in computer security*, pp. 77–101, Kluwer, 2002.

[36] D. E. Denning, "An Intrusion-Detection Model," *Software Engineering, IEEE Transactions on*, Vol. SE-13, No.2, pp. 222-232 ,1987.

[37] L. M'e and C. Michel, "Intrusion Detection," *A Bibliography, Sup'elec, Rennes*, Technical Report SSIR-2001-01, France, 2001

[38] X. Zhang, W. Wang, K. Nørvag and M. Sebag, "K-AP: Generating Specified K Clusters by Efficient Affinity Propagation," *Data Mining (ICDM), 2010 IEEE 10th International Conference*, pp. 1187 -1192, Washington, DC, USA, 2010.

[39] W. Barbakh and C. Fyfe. "Inverse weighted clustering algorithm," *Computing and Information Systems*, Vol. 11, No.2, pp. 10-18, 2007.

[40] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2 ed.)*, 2003

[41] J. Rennie, L. Shih, J. Teevan, and D. Karger, “Tackling the poor assumptions of Naive Bayes classifiers,” *In Proceedings of the Twentieth International Conference on Machine Learning*, Cambridge, 2003.

[42] L. Rokach, and O. Maimon, *Data mining with decision trees: theory and applications*, World Scientific Pub Co Inc, 2008.

[43] B. J. Frey, D. Dueck, “Clustering by Passing Messages Between Data Points,” in *Science*, vol. 315, No. 5814, pp. 972-976, 2007.

[44] Y. Jiay, J. Wangz, C. Zhangy, and X. Hua, “Finding Image Exemplars Using Fast Sparse Affinity Propagation,” *Proceedings of the 16th ACM International conference on Multimedia*, pp. 113 -118, New York, 2006.

[45] Y. Fujiwara, G. Irie, and T. Kitahara, “Fast Algorithm for Affinity Propagation,” *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2238-2243, California, 2011.

[46] C.-D. Wang, J.-H. Lai, C. Suen, and J.-Y. Zhu, “Multi-exemplar affinity propagation, Pattern Analysis and Machine Intelligence,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol.35, No. 9, pp. 2223–2237, 2013.

[47] K.J. Wang, J.Y. Zhang, D. Li, X.N. Zhang, and T. Guo, “Adaptive Affinity Propagation Clustering,” *Acta Automatica Sinica*, Vol. 33, No. 12, pp. 1242-1246, 2007.

[48] X. Liu, M. Yin, J. Luo and W. Chen, “An Improved Affinity Propagation Clustering Algorithm for Large-scale Data Sets,” *Ninth International Conference on Natural Computation (ICNC), IEEE*, pp. 894 – 899, Shenyang, 2013.

[49] Y. Bouzida, F. Cuppens, N. C. Boulahia, and S. Gombault, “Efficient intrusion detection using principal component analysis,” in *Proc. 3éme Conférence sur la Sécurité et Architectures Réseaux (SAR), La Londe, France*, 2004.

[50] Y. Bouzida, and F. Cuppens, “Neural networks vs. decision trees for intrusion detection,” in *Proc. IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM)*, pp. 28-29, Tuebingen, Germany, 2006.

[51] C.F. Tsai and C.Y. Lin, "A triangle area based nearest neighbor’s approach to intrusion detection", *Pattern Recognition*, Vol. 43, No.1, pp.222-229, 2010

[52] Z. Muda, W. Yassin, M.N. Sulaiman and N.I. Udzir, "Intrusion Detection based on K-Means Clustering and OneR Classification", in *7th International Conference on Information Assurance and Security (IAS)*, pp. 192-197, Melaka, Malaysia, 2011.

[53] W. Yassin, N.I. Udzir, Z. Muda and M.N. Sulaiman, "Anomaly-Based Intrusion Detection through K-Means Clustering and Naives Bayes Classification", *Proceedings of the 4th International Conference on Computing and Informatics (ICOCI),* pp. 298-303, Kedah, Malaysia, 2013.

[54] A. Purohit and H. Gupta, "Hybrid Intrusion Detection System Model Using Clustering, Classification and Decision Table", *Journal of Computer Engineering*, Vol. 9, No. 4, pp.103-107, 2013.

[55] V. Golmah, "An Efficient Hybrid Intrusion Detection System Based On C5.0 And SVM," *International Journal of Database Theory and Applications*, Vol.7 No.2, pp.59 – 70, 2014.

[56] M. Khosronejad and E. Sharififar, "Developing a Hybrid Method of Hidden Markov Models and C5.0," *International Journal of Database Theory and Application,* Vol.6, No.5, pp.165 – 174, 2013.

[57] T. Ghanem, W. Elkilani and H. Abdul-Kader, "A Hybrid approach for efficient anomaly detection using meta heuristic methods," *Journal of Advanced Research. Production and hosting by Elsevier B.V on half of Cairo University*, 2014.

[58] A. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading K-means clustering and C4.5 decision tree algorithm," *Procedia Engineering*, Vol.30, pp.174– 182,2012.

[59] W. Al-Yaseen, Z. Othman, and M. Nazri, "Hybrid Modified K-Means with C4.5 for Intrusion Detection Systems in Multiagent Systems," *The Scientific World Journal*, 2015.

[60] W. Gang, H. Jinxing, and M. Jian, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," *Expert systems with applications*, Vol.37, No.9, p.6225–6232, 2011.

[61] Y. Tsuruoka, and J. Tsujii, "Training a Naive Bayes Classifier via the EM Algorithm with a Class Distribution Constraint," *Proceedings of the seventh conference on Natural language learning at HLT-NAACL* –Vol. 4., Association for Computational Linguistics, Stroudsburg, 2003.

[62] D. Farid, N. Harbi, and M. Rahman, "Combining Naive Bayes and Decision Tree for adaptive intrusion detection," *International Journal of Network Security & Its Applications (IJNSA)*, Vol. 2, No 2, 2010.

[63] L. Cao, J. Zhong, and Y. Feng, "Construction Cosine RBF Neural Networks Based on Artificial Immune Networks," *ADMA'10 Proceedings of the 6th international conference on Advanced data mining and applications*, p.134-141, Berlin, 2010.

[64] T. Shaohua, D. Hongle, W. Naiqi, Z. Wei, and S. Jiangyi, “A Cooperative Network Intrusion Detection Based on Fuzzy SVMs,” *Journal of Networks*, Vol 5, No 4, pp.475–483, 2010.

[65] F. Amiri, Y. Mohammad, L. Caro, S. Azadeh, and Y. Nasser, “Mutual Information-Based Feature Selection for Intrusion Detection System,” *Journal of Network and Computer Applications*, Vol. 34, No.4, pp.1184–1199, 2011.

[66] S.J. Horng, “A novel intrusion detection system based on hierarchical clustering and support vector machines,” *Expert Systems with Applications*, Vol. 38, No. 1, pp. 306-313, 2011.

[67] N. Huy Anh, and C. Deokjai, “Application of Data Mining to Network Intrusion Detection: Classifier Selection Model,” *Lecture Notes in Computer Science*, Vol. 5297, pp. 399-408, 2008.

[68] G. Meera, and S.K. Srivatsa, “Classification Algorithms in Comparing Classifier Categories to Predict the Accuracy of the Network Intrusion Detection a Machine Learning Approach,” *Advances in Computational Sciences and Technology*, Vol. 3, No.1 pp.321–334, 2010.

[69] T. Mitchell, *Machine Learning*, McGraw-Hill Science, 1997.

[70] T. Korting, "C4.5 algorithm and Multivariate Decision Trees, Image Processing Division,” *National Institute for Space Research-INPE*, 2006.

[71] S. Siva, S. Sindhu, S. Geethab, and A. Kannan, “Decision tree based lightweight intrusion detection using a wrapper approach,” *Expert Systems with Applications: An International Journal*, Vol. 39, No.1, pp 129– 141, 2012

[72] J.H. Lee, J.H. Lee, and T. Chung “Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System,” Advanced Communication Technology, 10th International Conference on, Vol. 2, pp. 1170 – 1175, Gangwon-Do, 2008.

[73] D. Jachyra, K. Pancerz, and J. Gomula, “Classification of MMPI Profiles using Decision Trees,” *Concurrency, Specification and Programing, Poland*, pp 397-407, 2011.

[74] C.S. Trilok, J. Manoj, “WEKA Approach for Comparative Study of Classification Algorithm,” *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, No.4, pp 1925-1931, 2013

[75] D. J. Hand, and K. Yu, “Idiot's Bayes — not so stupid after all?” *International Statistical Review*, Vol. 69, No. 3, pp. 385–399, 2001.

[76] Z. Harry, and S. Jiang, “Naive Bayes for optimal ranking,” *Journal of Experimental and Theoretical Artificial Intelligence*, Vol. 20, No. 2, pp 79-93, 2008.

[77] C. L. Blake, C. J. Merz, "UCI repository of machine learning databases," 2012, http://archive.ics.uci.edu/ml/.

[78] S.T. Mai, He. Xiao, N. Hubig, C. Plant and C. Bohm, "Active Density-Based Clustering, Data Mining (ICDM)," *Data Mining (ICDM), IEEE 13th International Conference on*, pp 508–517, Dallas, TX , 2013.

[79] L. N. Ana, Fred, and K. J. Anil, "Robust Data Clustering," *Computer Vision and Pattern Recognition, 2003, Proceedings, 2003 IEEE Computer Society Conference*, Vol.2, pp. 128 – 133, Madison, WI, USA, 2003

[80] L. Koc, T. A. Mazzuchi, and S. Sarkani, "A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier," *Expert Syst. Appl*, Vol. 39, No. 18, pp. 13492–13500, 2012.

[81] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques (Google eBook)*, 2011.