International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Effective Impact of Intrusion Detection System for Manufacturing Industries Using Data Mining Techniques: A Comprehensive Study

Dr. T. A. Ashok Kumar

Professor, School of Science Studies, CMR University, Bengaluru. India
Email: *ashok1776[at]gmail.com*

Abstract: ***Intrusion detection systems are systems designed to monitor computer and network activities for security violations. These activities are observed by scrutinizing the audit data generated by the operating system or some other application programs running on the computer manufacturing industries have witnessed significant advancements in automation and connectivity, resulting in the proliferation of interconnected systems and devices. While this has increased efficiency and productivity, it has also exposed these industries to a growing risk of cyber-attacks and intrusions. Intrusion Detection Systems (IDS) play a crucial role in safeguarding manufacturing systems from unauthorized access and malicious activities. This paper presents an investigation into the use of data mining techniques to enhance the effectiveness of Intrusion Detection Systems in the context of manufacturing industries. Manufacturing environments are characterized by complex and heterogeneous data sources, making traditional rule-based IDS less effective in identifying novel and sophisticated attacks. Examples of security violations include the abuse of privileges or the use of attacks to exploit software or protocol vulnerabilities. Data mining techniques, such as machine learning algorithms and anomaly detection methods, offer the potential to address this challenge by learning from historical data and detecting previously unseen threats. This study explores the various data sources in manufacturing, including sensor data, network logs, and process data, and demonstrates how these can be integrated into a comprehensive IDS. The proposed approach is validated using real-world datasets and experimental results, showcasing its ability to effectively detect intrusions and anomalies in manufacturing systems. The integration of data mining techniques not only improves the accuracy of intrusion detection but also reduces false positives and enhances the adaptability of IDS to evolving threats.***

Keywords: Data Mining, Knowledge discovery, Intrusion systems, Propagation, Data Warehousing, ADAM, neural network, MADAM, firewall network, Intrusion Detection Systems, Manufacturing Treats, Cyber-Attacks, Cyber-Threats, Cyber Security, Clustering Algorithms.

1. Introduction

Traditionally, intrusion detection techniques are classified into two broad categories: *misuse detection* and *anomaly detection*. Misuse detection works by searching for the traces or patterns of well-known attacks. Clearly, only known attacks that leave characteristic traces can be detected that way. Anomaly detection, on the other hand, uses a model of normal user or system behavior and flags significant deviations from this model as potentially malicious. This model of normal user or system behavior is commonly known as the user or system profile. The strength of anomaly detection is its ability to detect previously unknown attacks.

Additionally, intrusion detection systems (IDSs) are categorized according to the kind of input information they analyze. This leads to the distinction between *Host-based* and *Network-based* IDS. Host-based IDSs analyze host-bound audit sources such as operating system audit trails, system logs, or application logs. Network-based IDSs analyze network packets that are captured on a network. Application-based intrusion detection uses the data obtained from application software such as web servers or some security devices. Many firewalls, access control systems, and other security devices generate their own event logs which contain information of security significance. *Target-based* intrusion detection doesn't require event data from any internal or external source. Instead this scheme provides a means of determining if the existing data in the system has been modified in some fashion. Target-based monitors use cryptographic hash functions to detect alterations to the system objects and then compare these alterations to some defined policy to detect any intrusion.

2. Scenario

The basic approaches in the Intrusion Detection System include rule-based systems, statistical analysis, and neural network approaches. All these systems described here focus on classification accuracy and none of them address the issue of learning time though some report large volumes of data to be a factor hindering in research. With the availability of microprocessors that perform billions of operations in a second and high-speed network connections, the size of the file recording all these events usually reaches in the order of gigabytes. Dealing with such a huge amount of data manually is not a trivial task and is not possible anymore. Specialized methods are needed to process its information contents. Data mining was viewed as a solution to this problem. Even mining large volumes of intrusion detection audit data requires a lot of computational time and resources. Traditional data mining algorithms are overwhelmed by the sheer complexity and bulkiness of the available data. They have become computationally expensive and their execution times largely depend on the size of the data they are dealing with to provide vital information about connections, most research uses secondary attributes also during the feature extraction.

Feature selection from the available data is vital to the effectiveness of the methods employed. Data mining

algorithms work more effectively if they have some amount of domain information available containing information on attributes that have higher priority than others, attributes that are not important at all, or established relationships that are already known. The most popular data format to do analysis on is the connection log. Besides being readily available and a much more reasonable size than other log formats (such as packet logs), the connection record format affords more power in the data analysis step, as it provides multiple fields where correlation can be done.

Data Mining:
Data mining is also known as Knowledge Discovery in Databases (KDD). Data mining involves the semiautomatic discovery of interesting knowledge, such as patterns, associations, changes, anomalies, and significant structures from large amounts of data stored in databases and other information repositories. Data mining is an information extraction activity whose goal is to discover hidden facts contained in databases. Using a combination of machine learning, statistical analysis, modeling techniques, and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results.

For every data mining system, a data preprocessing step is one of the most important aspects. Data preprocessing consumes 80% time of a Poor quality of data. The figure below explains the Architecture of Data Mining IDS consisting of sensors, detectors, a data warehouse, and a model generation component. This architecture is capable of supporting not only data gathering, sharing, and analysis.

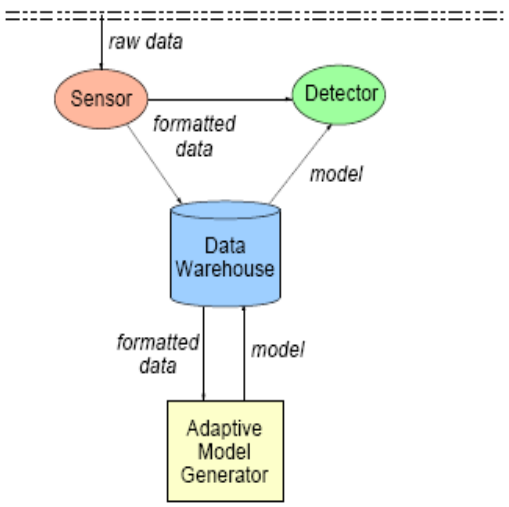

Figure 1

Sensors:
Sensors observe raw data on a monitored system and compute features for use in model evaluation. Sensors insulate the rest of the IDS from the specific low-level properties of the target system being monitored. This is done by having all the sensors implement a Basic Auditing Module (BAM) framework.

Detectors:
Detectors take processed data from sensors and use a detection model to evaluate the data and determine if it is an attack. The detectors also send back the result to the data warehouse for further analysis and report. There can be several (or multiple layers of) detectors monitoring the same system. For example, workloads can be distributed to different detectors to analyze events in parallel. There can also be a "back-end" detector, which employs very sophisticated models for correlation or trend analysis, and several "front-end" detectors that perform quick and simple intrusion detection. The front-end detectors keep up with high-speed and high-volume traffic and must pass data to the back-end detector to perform more thorough and time-consuming analysis.

Data Warehouse:
The data warehouse serves as a centralized storage for data and models. One advantage of a centralized repository for the data is that different components can manipulate the same piece of data asynchronously with the existence of a database, such as offline training and manual labeling. The same type of components, such as multiple sensors, can manipulate data concurrently. The relational database features support "stored procedure calls" which enable easy implementation of complicated calculations, such as efficient data sampling carried out automatically on the server. The data warehouse also facilitates the integration of data from multiple sensors. By correlating data/results from different IDSs or data collected over a longer period of time, the detection of complicated and large-scale attacks becomes possible.

Adaptive Model Generator:
The main purpose of the model generator is to facilitate the rapid development and distribution of new (or updated) intrusion detection models. In this architecture, an attack detected first as an anomaly may have its exemplary data processed by the model generator, which in turn, using the archived (historical) normal and intrusion data sets from the data warehouse, automatically generates a model that can detect the new intrusion and distributes it to the detectors (or any other IDSs that may use these models). Especially useful are unsupervised anomaly detection algorithms because they can operate on unlabeled data that can be directly collected by the sensors.

Proposed Algorithms:
MADAM ID – Mining Audit Data for Automated Models for Intrusion Detection. MADAM ID is a network-based intrusion detection system that uses a data mining approach to detect anomalies as well as misuse detection. The main components of MADAM ID are classification and meta-classification programs, association rules and frequent episodes programs, a feature construction system, and a conversion system that translates off-line learned rules into real-time modules. Using MADAMID, raw audit data is first preprocessed into records with a set of "intrinsic" (i.e., general purposes) features, e.g., duration, source and destination hosts and ports, number of bytes transmitted, etc. Data mining algorithms are then applied to compute the frequent activity patterns, in the forms of association rules and frequent episodes, from the audit records. Association

rules describe correlations among system features Together, association rules and frequent episodes form the statistical summaries of system activities. Domain knowledge is required in MADAM ID. Human experts need to first define a basic set of features as the seed for the automatic feature construction process.

ADAM – Audit Data Analysis and Mining

ADAM is a real-time network-based anomaly detection system. It employs data mining to extract association rules from the audit data. ADAM works by creating a customizable profile of rules of normal behavior and it contains a classifier that distinguishes the suspicious activities, classifying them into real attacks and false alarms.

ADAM is essentially a test bed for using data mining techniques to detect intrusions. ADAM uses a combination of association rules mining and classification to discover attacks in a TCP dump audit trail. First, ADAM builds a repository of "normal" frequent item sets that hold during attack-free periods. It does so by mining data that is known to be free of attacks. Secondly, ADAM runs a sliding-window, online algorithm that finds frequent item sets in the last D connections and compares them with those stored in the normal item-set repository, discarding those that are deemed normal. With the rest, ADAM uses a classifier that has been previously trained to classify the suspicious connections as a known type of attack, an unknown type, or a false alarm.

3. Implementation details of the Algorithm:

ADAM is unique in two ways.

First, ADAM uses data mining to build a customizable profile of rules of normal behavior, and a classifier that sifts the suspicious activities, classifying them into real attacks (by name) and false alarms.

Secondly, ADAM is designed to be used online (in real-time), a characteristic achieved by using incremental mining algorithms that use a sliding window of time to find suspicious events.

ADAM uses a combination of association rules mining and classification to discover attacks in a TCP dump audit trail. First, ADAM builds a repository of "normal" frequent item sets that hold during attack-free periods. It does so by mining data that is known to be free of attacks. Secondly, ADAM runs a sliding-window, online algorithm that finds frequent item sets in the last D connections and compares them with those stored in the normal item-set repository, discarding those that are deemed normal. With the rest, ADAM uses a classifier that has been previously trained to classify suspicious connections as a known type of attack, an unknown type, or a false alarm.

In this schema, Ts represents the beginning time of a connection, Src: IP and refers to the source IP and port number respectively, while Dst: IP and Dst: Port, represent the destination IP and port number. The attribute FLAG describes the status of a TCP connection. The relation R contains the dataset that is the subject of the association mining.

The number of potential itemsets is large: connections may come from a large base of source IP addresses and ports. We focus on item sets that contain items that indicate the source of the connection (like source IP and port), and items that indicate its destination (like destination IP and port). We also consider itemsets that are “aggregations" of source IP or Port values, e.g., connections that come from a source domain and have the same destination IP. We call these itemsets domain-level itemsets. (For instance, we want to discover frequent connections from Source IP X to Destination IP Y or from Source Domain W to Destination IP Y.)

First, ADAM is trained using a data set in which the attacks and the attack-free periods are correctly labeled. In the first step, a database of frequent item sets (those that have support above a certain threshold) for the attack-free portions of the data set is created. This serves as a profile against which frequent itemsets found later will be compared. The profile database is populated with frequent itemsets whose format was shown before, as well as frequent domain-level itemsets for attack-free portions of the data.

Next, to complete the training phase, we use an incremental, on-line algorithm to detect itemsets that receive strong support within a period of time. This algorithm is driven by a sliding window of tunable size ±. The algorithm outputs itemsets (of the same format of those present in the profile database) that have received strong support during this window. We compare any itemset that starts receiving support with itemsets in the profile database for an analogous time and day of the week. If the itemset is present in the profile database, we do not pay attention to it (i.e., we do not devote storage resources to keep track of its support). On the other hand, if the itemset is not in the database, we keep a counter that will track the support that the item receives. If the itemset's support surpasses a threshold, that item is reported as suspicious.

As we know where the attacks are in the training set, the corresponding suspicious itemsets along with their feature vectors are used to train a classifier. The trained classifier will be able to, given a suspicious itemset and a vector of features, classify it as a known attack (and label it with the name of the attack), as an unknown attack (whose name is not known), or as a false alarm.

It is important to remark here that ADAM has the ability to classify a suspicious event (item and features) as an unknown attack. Notice that no training set can possibly prepare a classifier for an unknown attack (since there can be no examples of such an event). In general, labeling events as unknown attacks (or anomalies) is a very difficult problem.

ADAM is then ready to detect intrusions online. Again, the online association rules mining algorithm is used to process a window of the current connections. Suspicious connections are tagged and sent along with their feature vectors to the trained classifier, where they will be labeled accordingly.

The most common type of genetic algorithm works in the following steps:

1) A population is created with a group of individuals created randomly.
2) The individuals in the population are then evaluated.
3) The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task.
4) Two individuals are then selected based on their fitness, the higher the fitness, the higher and the chance of being selected. These individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly.

This process continues until a suitable solution has been found or a certain number of generations have passed, depending on the needs of the programmer.

A. Selection

There are many different types of selection; the most common type is called the roulette wheel selection. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are then chosen randomly based on these probabilities and produce offspring. Pseudo-code for a roulette wheel selection algorithm is shown below:

Classical clustering algorithms generate a partition of the population in a way that each case is assigned to a cluster. These algorithms use the so-called ‘‘rigid partition’’ derived from the classical sets theory: the elements of the partition matrix obtained from the data matrix can only contain values 0 or 1; with zero indicating null membership and one indicating whole membership. Fuzzy partition is a generalization of the previous one, so that it holds the same conditions and restraints for its elements, except that in this case real values between zero and one are allowed (partial membership grade). Therefore, samples may belong to more than one group.

```
For all members of the population
   sum += fitness of this individual
end for

For all members of the population
   Probability = sum of probabilities + (fitness / sum)
   the sum of probabilities += probability
end for

loop until the new population is full
    do this twice
       number = Random between 0 and 1
      for all members of the population
         If number > probability but less than the next probability
              then you have been selected
      end for
    end
    create offspring
end loop
```

B. Crossover

Crossover operates by selecting a random location in the genetic string of the parents (crossover point) and concatenating the initial part of one parent with the final part of the second parent to create a new child. A second child is simultaneously created by the remaining parts of the two parents. Different types of crossover (single point, Two points, Uniform, etc), the most common type is single point crossover. In single-point crossover, a locus is chosen at which it swaps the remaining alleles from one parent to the other. This is complex and is best understood visually.

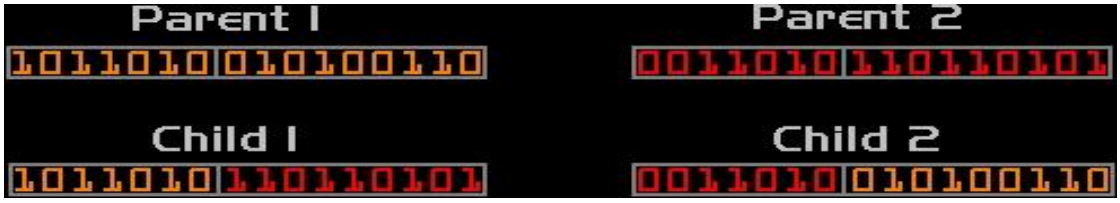

Figure 2: Crossover of data bits

Here, the children take one section of the chromosome from each parent. The point at which the chromosome is broken depends on the randomly selected crossover point. This particular method is called single-point crossover because only one crossover point exists. Sometimes only child 1 or child 2 is created, but oftentimes both offspring are created and put into the new population. Crossover does not always occur, however. Sometimes, based on a set probability, no crossover occurs and the parents are copied directly to the new population. The probability of crossover occurring is usually 60% to 70%.

Basic K-Means Algorithm

Cluster Analysis is the problem of decomposing or partitioning a (usually multivariate) data set into groups so that the points in one group are similar to each other and are as different as possible from the points in other groups. There are many situations where clustering can lead to the discovery of important knowledge but privacy/security reasons restrict the sharing of data. Imagine the following scenario.

A law enforcement agency wants to cluster individuals based on their financial transactions, and study the differences between the clusters and known money laundering operations. Knowing the differences and similarities between normal individuals and known money launderers would enable better direction of investigations. Currently, an individual's _financial transactions may be divided between banks, credit card companies, tax collection agencies, etc. Each of these (presumably) has erective controls governing the release of the information.

These controls are not perfect, but violating them (either technologically or through insider misuse) reveals only a subset of an individual's _financial records. The law enforcement agency could promise to provide erective controls, but now overcoming those gives access to an individual's entire financial history. This raises justifiable concerns among privacy advocates. What is required is a privacy-preserving way of doing clustering.

K-means is an iterative clustering algorithm in which items are moved among a set of clusters until the desired set is reached.[12] The operation of K-means is illustrated in Figure (2.6), which shows how, starting from three centroids the final cluster is found in four assignment update steps. In these and other figures displaying K-means clustering each subfigure shows(1) the centroids at the start of the iteration and (2) the assignment of the point to those centroids. The centroids are indicated by the "+" symbol; all points belonging to the same cluster have the same marker shape.

Algorithm:

1) Select K point as initial centroids.
2) Repeat
3) From the K cluster by assigning each point to its closest centroid
4) Until Centroids do not change.

C. Mutation

After selection and crossover, now have a new population full of individuals. Some are directly copied, and others are produced by crossover. In order to ensure that the individuals are not all exactly the same, it's possible to allow for a small chance of mutation.

Figure 3: Crossover of data bits

It loops through all the alleles of all the individuals, and if that allele is selected for mutation, then can either change it by a small amount or replace it with a new value. The probability of mutation is usually between 1 and 2 tenths of a percent.

4. Conclusion

We have implemented feature extraction and construction algorithms for labeled audit data (i.e. when both normal and intrusion data sets are given). We are implementing algorithms for unlabeled data (which can be purely normal or possibly contain unknown intrusions).In summary, data mining is proving to play an important role in intrusion detection. We are very confident that ADAM will become, with time, a very strong tool to help security officers in their daily work against intruders. In conclusion, the implementation of Intrusion Detection Systems (IDS) using data mining techniques represents a significant advancement in the field of cyber security. The utilization of data mining algorithms and methodologies in IDS has demonstrated a range of benefits and capabilities that enhance the overall effectiveness of intrusion detection.

References

[1] Osmar R. Zaïane, "Chapter I: Introduction to Data Mining", CMPUT690 Principles of Knowledge Discovery in Databases, 1999.
[2] M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.
[3] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, "Advances in Knowledge Discovery and Data Mining", AAAI/MIT Press, 1996.
[4] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
[5] Lauri Tuovinen, Perttu Laurinen, Ilmari Jutilainen, and Juha Roning, "Data mining applications for diverse industrial Application domains with smart archive", Proceedings of the IASTED International Conference on Software Engineering, Pages 56-61, 2008.
[6] Cheng-Fa Tsai; Han-Chang Wu; Chun-Wei Tsai, "A new data clustering approach for data mining in large databases", Proceedings. International Symposium on Parallel Architectures, Algorithms and Networks, 2002. I-SPAN '09.
[7] Wenke Lee, Salvatore J. Stolfo, Philip K. Chan, Eleazar Eskin, Wei Fan, Matthew Miller, Shlomo Hershkop, and Junxin Zhang "Real-Time Data Mining-based Intrusion Detection", 2011.
[8] Tzu-Yi Yang and Fang-Yie Leu, "A host-based real-time intrusion detection system with data mining and forensic techniques", Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, 2003.
[9] Chi-Ho Tsang and Kwong, S., "Multi-agent intrusion detection system in the industrial network using ant colony clustering approach and unsupervised feature extraction", IEEE International Conference on Industrial Technology, 2005. ICIT 2008.
[10] Chin Yuan Fan, Lai, M.F. Huang, T.Y.; Huang, C.M., "Applying K-means clustering and technology map in Asia Pacific-semiconductors industry analysis", IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2011.