Proposal of Honeypot-Based Data Mining Methods for the Discovery of Intrusions in Big Data Databases

Koffi Kanga[1(✉)], Beman Hamidja Kamagaté[2], Raogo Kabore[3], and Souleymane Oumtanaga[4]

[1] ESATIC (Ecole Supérieure Africaine des TIC: Republic of Côte d'Ivoire), Abidjan, Côte d'Ivoire
Kanga.koffi@esatic.edu.ci

[2] Laboratory of Information, Communication Sciences and Technologies, (Ecole Supérieure Africaine Des TIC), LASTIC-ESATIC, Abidjan, Cote d'Ivoire, 18bp, 1501 Abidjan, Côte d'Ivoire
beman.kamagate@esatic.edu.ci

[3] Communication Sciences and Technologies (Ecole Supérieure Africaine Des TIC), LASTIC-ESATIC, Abidjan, Cote d'Ivoire, 18bp, 1501 Abidjan, Côte d'Ivoire
raogo.kabore@esatic.edu.ci

[4] Computer Science and Telecommunications Research Laboratory (Institut Nationale Polytechnique Houphouet Boigny), LARIT - INPHB, Yamoussoukro, Côte d'Ivoire

Summary. In this paper we propose a data mining technique for the discovery of intrusions in big data. To achieve our objective, we first reviewed the different data mining works and tools to our knowledge for the extraction of data from big data. Secondly, we chose a honeypot (honeyD) from a set (of honeypots) based on well-defined criteria. Thirdly, we combined this honeypot (honeyD) with different classification algorithms (decision trees and clustering such as k-means, DBSCAN to identify possible intrusions into the databases) in a functional architecture in which, we have presented and explained the role of each of its components. The implementation of our proposal shows that the combination of the honeypot with these different clustering algorithms gives convincing results which make it possible to detect possible intrusions in the data big databases.

Keywords: Intrusion detection · computer security · data mining · big data

1 Introduction

Talking about data mining methods for detecting intrusions in big data databases using honeypots deserves explanation.

Indeed, data mining is a set of techniques allowing extracting data or knowledge in the form of models allowing to describe the current behavior and/or to predict the future behavior of the system [2]. To do this, data mining makes use of statistical techniques, databases, data analysis and artificial intelligence.

A. Sere et al. (Eds.): AFRICOMM 2023, LNICST 587, pp. 312–332, 2025.
https://doi.org/10.1007/978-3-031-81570-6_21

As for big data, it is a concept that became popular in 2012 to reflect to the fact that companies are faced with increasingly large volumes of data to process, which present a strong commercial and marketing challenge. Several definitions exist in the literature [3], but we retained the one which stipulates that it is a set of technologies, architectures, tools and procedures allowing an organization to very quickly capture, process, analyze large quantities and heterogeneous and changing content, then extract relevant information at an affordable cost.

For the storage of this variety of data, big data makes use of four (4) families of DBMS called NoSQL (Key-value oriented database, document-oriented database, column-oriented database and graph-oriented database).

As for a honeypot, it would be defined according to the context of use. Thus, in the literature honeypots are defined as a means of attracting attackers, while others consider them as tools to detect intrusions.

In [6], a honeypot is defined as an effective counter -measure to prevent unauthorized use of critical information systems in networks. In the remainder of our paper, we adopt the following definition: "A honeypot is a secure resource which is set up and which has the objective of attracting hackers with the aim of not attacking or compromising them"

Furthermore, an intrusion is any use of a computer system for purposes other than those intended, generally due to the acquisition of privileges in an illegitimate manner.

For intrusion detection, it consists of analyzing the information collected in search of possible attacks by security audit mechanisms.

As for an intrusion detection system, it would be a set of tools and methods used to detect and report abnormal activities produced in a computer system. Thus, an IDS aims to protect a system from malicious activities coming from known or unknown sources. This protection is provided automatically to ensure the confidentiality, integrity and availability of the systems. Cannady et al. I in [7] states that an IDS has two detection approaches: anomaly-based detection and signature-based detection [7].

Today, the digital revolution with its corollary of exponential data growth, capturing these large volumes of data from various sources to be processed at a high and acceptable speed would be a wish; but securing this data seems even better. However, this security requires the implementation of a set of tools and processing methods (MAPREDUCE algorithm, Machine Learning, Deep Learning, etc.) based on data mining techniques in the big data databases.

The remainder of our paper is organized as follows:

- Sect. 2, we present the state of the art
- Sect. 3, we identify our problem
- Sect. 4, we illustrate our contribution and in
- Sect. 5, we present an implementation of our contribution
- Sect. 6 is devoted to a discussion and we will end with a conclusion in Sect. 7. In this section, we will identify some perspectives

2 State of the Art

2.1 Intrusion detection

In a computer system, intrusion detection involves two (2) approaches. The first is to look for signatures of known attacks while the second is to define normal system behavior and look for what does not fit into that behavior.

2.1.1 Signature Approach [8]

The signature approach consists in defining attack scenarios and searching for traces of these scenarios in the system. This system could contain big data databases and system audit (log) files.

- **The search for patterns**: this IDS-based method makes use of data from a database containing a set of signatures. In this set, each signature contains information about the protocols and ports used by a specific attack and a pattern to recognize suspicious packets.
- **Generic search**: it is suitable in the case of virus signatures. We look in the executable code for commands that are potentially dangerous, such as unreferenced and detected DOS commands, email broadcasts, instructions linked to known attacks.
- **Protocol analysis**: this is a method based on a flow conformity check and an observation of suspicious fields and parameters. This approach makes it possible to detect unknown attacks.
- **Integrity Check**: it takes a photo of all the files on a system and generates an alert in the event of corruption of one of the files.
- **Heuristic analysis and** anomaly detection: this approach carries out intelligent analysis which facilitates the detection of suspicious activity or any other anomaly.

2.1.2 Behavioral Approach [8]

The basic principle of the behavioral approach is to construct a reference model of the behavior of the monitored entity (user, machine, service, application) to which the observed behavior can be compared. If the latter is too far from the reference, an alert is issued to report the anomaly.

- **Probabilistic analysis**: this approach is sometimes described as **Bayesian**: Bayesian networks make it possible to model situations in which causality plays a role, but where knowledge of all the relationships between phenomena is incomplete, so that it is necessary to describe them in a probabilistic way. Thus, for each element of the profile, the probability of each event likely to occur subsequently is specified. The indications obtained progressively on the state of the modeled system influence the confidence granted to a given proposition.
- **Statistical analysis**: in this approach, the profile is established by observing the value of certain parameters of the system considered as random variables. For each system parameter, a statistical model is used to establish the distribution of the corresponding random variable. Once the model is established, a distance vector is calculated between the stream of observed events and the profile. If the distance exceeds a certain threshold, an alert is issued.

- **Immunology**: builds a model of normal behavior of services (and not users). Here we observe a service for a long enough time in good conditions to build a complete behavior model.
- **graphs**: The goal is to highlight properties and the relationships between these properties. The advantage of this approach is that it makes it easier to process rare events.

2.2 Architecture and Intrusion Detection Honeypot

2.2.1 Intrusion Detection Architecture [9]

Intrusion detection architectures can be classified into three categories:

- **Centralized architecture:** In this architecture, all intrusion detection data is collected and analyzed from a centralized location. This can include data from different endpoints such as firewalls, IDSs, IPSs, anti-virus software. This architecture is generally used in fixed networks and enterprises.
- **Distributed and cooperative architecture:** In this architecture, the data is shared between the nodes for a global analysis of the security of the network. Alerts are generated if an intrusion is detected by multiple nodes. This architecture is generally used in wireless sensor networks and ad hoc networks.
- **Hierarchical architecture:** In this architecture, certain nodes are designated as monitoring nodes to collect and analyze intrusion detection data. In Table 1 below, we make a comparison of these architectures.

Table 1. Comparative table of IDS architectures

Comparison criterion	Centralized architecture	Distributed and cooperative architecture	Hierarchical architecture
Global overview	Yes	Yes	Yes
Centralized alert management	Yes	No	Yes
Ability to detect threats	Yes	Yes	Yes
Single point of failure	Yes	No	Yes
Reliability	Yes	Medium to high	Medium to high

2.2.2 Honeypots for Intrusion Detection [9]

In [9], the authors present the honeypots according to their implementation environment and their level of interaction. So, for the implementation environment, they present the

production and research honeypot. The production honeypots are used to detect attacks from the outside while the research honeypots are used to study the activities of hackers.

According to the level of interaction, the authors present:

– **Low Interaction Honeypots**

They would be easier to install, configure, deploy and maintain due to their fairly simple design. They are best known for their powers of detecting unauthorized connections. Since the functionalities offered by these honeypots are limited, the level of attack risk also seems limited. This type of honeypot is not designed to discover new attacks, but rather to monitor and analyze the network environment of the place where they are installed.

– **Medium Interaction Honeypots**

These honeypots offer more functionality than those in the previous category and therefore their interaction levels are relatively higher. They require more attention during installation and configuration. However, the configuration must always take into consideration network security when the honeypot is attacked.

– **High Interaction Honeypots**

These honeypots provide more information about attacks, but consume more time during installation, configuration and maintenance. They also present a very high level of risk because they give attackers greater control over the operating system. Moreover, these honeypots are often installed in an uncontrolled portion of the network (e.g. absence of a firewall) and it is often necessary to strengthen the security of the rest of the network in order to minimize the risk that the honeypot be used as a starting point for an attack.

2.2.3 Data Mining Methods [10]

We distinguish two Data mining techniques:

Techniques Supervised
They produce prediction models, which from the values of a set of predictor variables (input values), predict the value of a target variable or variable to be explained (output value). They include three classes of techniques:

- Estimation: brings together the techniques which make it possible to define the link between a set of predictor variables and a target variable of numerical type.
- Classification: brings together the techniques which make it possible to define the link between a set of predictor variables and a categorical target variable, most often Boolean.
- Forecasting: similar to estimation and classification except that the results relate to the future.

Unsupervised Techniques
They produce clustering models, which from the values of a set of variables, they classify the current object in a class (cluster), the classes are unknown in advance. In this category, there are three classes of techniques:

- Description: gathers the techniques which make it possible to describe the links between the various variables of the concept.
- Grouping (clustering): groups the techniques which make it possible to create classes of data similar between them and different from the data of another class (that is to say, the intersection between the classes must always be empty).
- Association: gathers the techniques which make it possible to describe the links between the values of the various variables of the concept by producing for example a model of rules of association.

2.2.4 Algorithm Types

– Association rule detection algorithm: This algorithm uses data mining techniques to identify association rules between system activities and intrusions. It can detect intrusions that have similar patterns to previous intrusions, but it may be sensitive to variations in data and false alarms.
– Behavior Detection Algorithm: This algorithm uses behavior monitoring techniques to identify abnormal user and system behavior. It can detect intrusions that have never been seen before, but it can be sensitive to variations in normal behavior and false alarms.
– Correlation algorithms: use correlation techniques to detect relationships between different security events and identify potential attacks.
– Model-based algorithms: build a model of normal behavior and detect anomalies by comparing real data to this model.
– Neural network-based detection algorithm: This algorithm uses neural networks to identify anomalous activity in the data (Table 2).

Table 2. Comparative table of data mining algorithms

Big data database family	Association rule detection	Behavior detection	Correlation detection	Detection based on neural networks	Model-based detection
Key-value	No	Yes	No	Yes	No
Document oriented	No	Yes	No	Yes	Yes
Column oriented	Yes	Yes	Yes	Yes	Yes
Graphs	No	No	Yes	Yes	No

2.3 Intrusion Detection Tools [10]

In the literature, several intrusion detection tools exist. In this part we present the most representative ones to our knowledge in terms of their mode of operation and their different architectures.

2.3.1 SNORT

SNORT is an Open Source Network Intrusion Detection System (NIDS), capable of analyzing real-time traffic on IP networks.

SNORT is able to perform real-time network traffic analysis and is equipped with different intrusion detection technologies such as protocol analysis and pattern matching, it can detect many types of attacks such as: malware, buffer overflows, port scans and sniffing.

- Operating Mode

The "offline" sniffer mode which simply reads the packets circulating on the network and displays them continuously on the screen. It is a question of listening to the network, by typing one or more lines of commands which will indicate to SNORT the type of result to be displayed.

The "packet logger" mode which logs packets to disk. This mode is in all respects similar to the previous one, except that the logs are no longer displayed on the screen, but are entered directly into a log file.

The more configurable NIDS mode, which allows to analyze the traffic on the network following rules defined by the user and to establish actions to be carried out according to the cases.

- SNORT Architecture [11, 12]

The essential components of the SNORT architecture are:

– **Packet Decoder**: it captures data packets from network interfaces, prepares them to be pre-processed or sent to the detection engine.
– **Pre-processor**: these are components used with SNORT to improve the possibilities of analysis and recomposition of captured traffic. They receive the packets, reprocess them and send them to the detection engine.
– **Detection Engine**: This is the most important component of SNORT. Its role is to detect any intrusions that exist in a packet. To do so, the search engine is based on the rules of SNORT. Indeed, this engine consults these rules and compares them one by one with the data packet. If there is compliance, the detector records it in the log file and/or generates an alert. Otherwise the packet is dropped.
– **Logging and Alerting System:** it allows to generate alerts and log messages according to what the detection engine has found in the analyzed packet.
– **Output modules** (or plugins): allows the intrusion generated by the alert and notification system to be processed in several ways (sends to a log file, generates an alert message to a Syslog server, or stores this intrusion in a database) (Fig. 1).

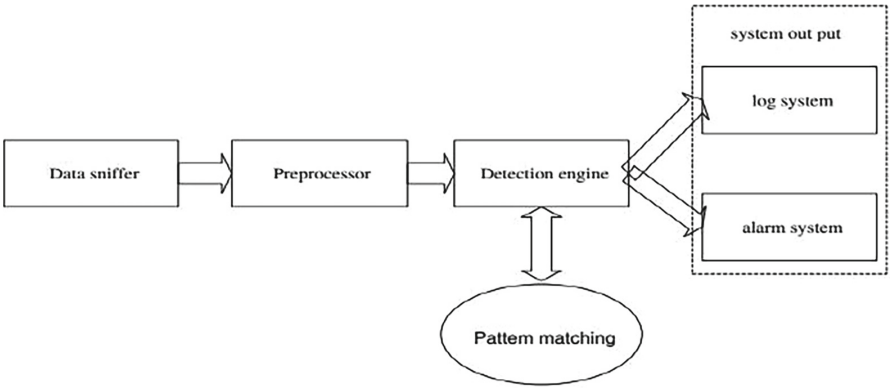

Fig. 1. Simplified architecture of SNORT [11]

2.3.2 Meerkat [13]

Suricata is an open-source intrusion detection system (IDS) that helps detect attacks and anomalies on computer networks. It is capable of processing high-speed data streams using hardware or software acceleration technologies to improve performance. It is based on a multi-threaded architecture and uses detection rules written in Lua language. It can be used in detection, prevention or network analysis mode. It supports a variety of network protocols, such as TCP, UDP, HTTP, DNS and SLL, it can detect anomalies and suspicious behavior in the data of these protocols. It uses different detection mechanisms, including basic signatures, behavior analysis, and protocol rules. It can also use network analysis tools to identify trends and attack patterns. It can integrate with existing security tools such as Security Management Systems (SIEM) for comprehensive analysis.

- **Operating Mode**

Suricata's mode of operation is based on a layered architecture that enables efficient processing of network data. It consists of several layers, each playing a specific role in detecting intrusions.

– **Packet capture**: The first layer is responsible for capturing network packets using technologies such as LIB PCAP or PF_RING. The packets are then forwarded to the next layer for analysis.
– **Protocol analysis**: The second layer is responsible for analyzing network protocols such as TCP, UDP, HTTP, DNS and SSL. It uses mechanisms such as base signatures, behavior analysis, and protocol rules to detect anomalies and suspicious behavior.
– **Intrusion detection**: The third layer is responsible for detecting intrusions using the information collected by the previous layers. It uses network analysis tools to identify trends and attack patterns.
– **Notification**: The last layer is responsible for the notification of alerts and detected incidents. Alerts can be sent by email, SMS or via a web interface. It can also integrate with existing security tools such as Security Management Systems (SIEM) for comprehensive analysis (Fig. 2).

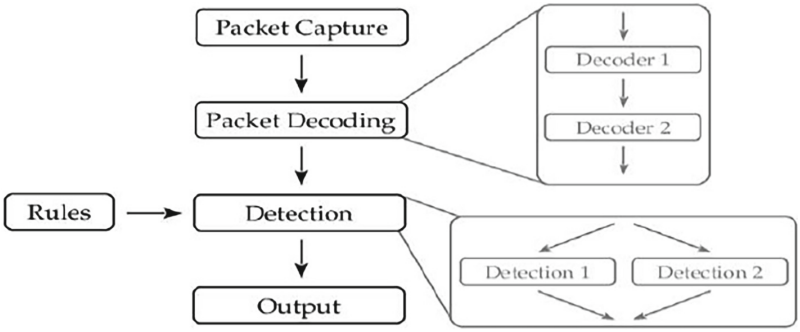

Fig. 2. SURICATA architecture

2.3.3 OSSEC (OpenSource Security)

It is an open-source intrusion detection system (HIDS) that helps monitor activities on operating systems and networks. It uses signature-based detection techniques to detect known intrusions and behavior-based detection to detect unknown intrusions. It can also use log analysis tools to identify trends and attack patterns. It is based on a client-server architecture and makes it possible to centralize the security data of several systems; which facilitates the management of alerts and decision-making. It supports a large number of operating systems, such as Windows, Linux, MacOs, and BSD (Fig. 3).

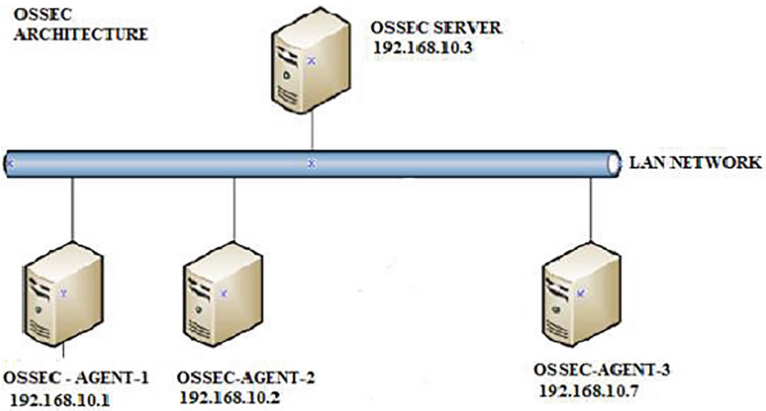

Fig. 3. Architecture of OSSEC [14]

- **Operating Mode**

OSSEC's mode of operation is based on client-server architecture. It consists of several components that work together to detect intrusions and anomalies on systems and networks. These are:

- **Agent**: this component consists of agents installed on the systems and networks to be monitored. They collect security information, such as event logs and security alerts, and send them to the OSSEC server for analysis.
- **Server**: The **server component** is responsible for analyzing the data collected by the agents. It uses signature-based detection rules and machine learning algorithms to identify anomalies and suspicious behavior. It generates alerts in the event of intrusion detection.
- **Console**: The OSSEC **console component** is a web interface that allows you to manage alerts, rules and configurations. It also allows you to view analytics data and generate reports.
- **Notification**: this component of OSSEC is able to notify administrators in the event of intrusion or anomaly detection via email or SMS notifications. It can also integrate existing security tools such as security management systems (SIEM) for comprehensive analysis.

2.4 Other Intrusion Detection Tools (IDS)

It is important to note that this list is not exhaustive and that new software is developed regularly, so it is important to keep researching the latest technology to ensure that you are using the best tool. For these other forms of tools, we classify them into 3 (three) categories. These are:

- Network Based Tools (NIDS): Bro, Tripwire, McAfee Network Security Platform, Symantec DeepSight IDS, Prelude-IDS, SELKS
- Host Based Tools (HIDS): McAfee Host Intrusion Prevention System, Wazuh, AppArmor, SELinux, ClamAV
- IDS hybrid tools: OSSEC, AIDE, Tripwire

Table 3. Advantages and disadvantages of other IDSs

	Benefits	Disadvantages
NESTS	Alarm in case of anomaly Multiple positioning Real Time Execution	Signatures to be updated, Traffic absorption. Inoperative for encrypted streams Management of false positives, Expertise desired
HIDS	Station protection Real-Time Execution	Ineffective against attacks on multiple hosts. Different configurations depending on the systems used
Hybrid	Reduction of false positives Real Time Execution Ensures the correlation of events	More sources, more difficult management and interpretation of alarms

For comparison, we present the advantages and disadvantages of these 3 categories in Table 3.

3 Problematic

From this literature review, it appears that excellent research work has been carried out. This work resulted in the implementation or use of IDS. Others use data mining techniques and methods to highlight malicious activity. The question that arises at the moment would be to know what result would we obtain if we proceeded to set up an environment using data mining tools coupled with honeypots?

4 Contribution

In this part, we make the choice of honeypot based on well-defined criteria. This choice will be coupled with a data mining technique which is itself selected according to equally defined criteria. Thus presented, our contribution is the combination of a honeypot with different data mining techniques in an architecture whose components and roles will be presented.

4.1 Choosing the Type of Honeypot

4.1.1 Selection Criteria

In order to choose the type of honeypot best suited to our needs, we define the selection criteria according to the objectives set. To do this, we have:

- **The relevance of the data collected**: The honeypot must be able to collect the data relevant to our analysis.
- **Event log collection**: The honeypot must be able to collect real-time event logs for further analysis.
- **Scalability**: In the case of big data, it is important to choose a honeypot that can evolve according to the growth of the data.
- **Data security**: It is important to ensure that the selected honeypot can protect collected data and event logs from unauthorized access.
- **Big data database configuration**: A honeypot that can be configured to connect and collect data from big data databases.

By taking into account these aforementioned selection criteria, we will be able to make an informed choice for the type of honeypot that will best suit our needs.

4.1.2 Evaluation of the Different Types of Honeypots

The identified honeypots are classified according to the level of interaction: low, medium and high.

The Relevance of the Data Collected: In general, honeypots with a high level of interaction are considered to be the most suitable in terms of the relevance of the data collected. This is due to their ability to more realistically simulate target environments for attackers. Honeypots with a lower level of interaction can also be used to collect useful data, but they may be less efficient in terms of the relevance of the data collected.

Event Log Collection: Highly interacting honeypots are best suited for event log collection. They can capture and record a large amount of information about malicious interactions with the system. Low interaction level and medium interaction level honeypots also collect event logs, but less in a comprehensive and detailed way.

Scalability: Low-level interaction honeypots are the most suitable for scalability, as they are generally simpler and lighter to manage. High-interaction honeypots can be more difficult to manage and use due to their increased complexity. Medium-level interaction honeypots fall somewhere in between in terms of ease of management and use.

Data Security: When it comes to security, high interaction level honeypots are considered the most secure, followed by medium interaction level honeypots and finally low interaction level honeypots. High interaction honeypots are considered the most secure because they mimic a real system or application hence the attention required for configuring and securing the system. Unlike the other types of honeypot mentioned.

Big Data Database Configuration: Honeypots with a high level of interaction are the most suitable for configuring big data databases. They offer large data storage and processing capacity. Honeypots with low levels of interaction are generally considered the least effective, as they lack the functionality needed to process the enormous amounts of data. As for honeypots with a medium level of interaction, they generally have a greater variety of functionalities for processing data.

4.1.3 Choice and Justification

Table 4. Summary of the strengths and weaknesses of the different types of honeypots according to the criteria

Criteria	Low Level Honeypot	Mid -Level Honeypot	High Level Honeypot
Relevance of the data collected	Low to medium	Medium to high	High
Collecting event logs	Medium to high	High	High
Scalability	High	Medium to high	Medium to high
Data security	Average	High	High
Configuring the big data database	Low to medium	Average	High

As established in the table, high-level honeypots best meet the various criteria set out. Our choice then fell towards a honeypot with a high level of interaction (HoneyD) (Table 4).

4.2 Choice of Data Mining Methods

4.2.1 Selection Criteria

In order to choose the data mining method best suited to our needs, it is important to define the selection criteria according to the objectives set. To do this, we have:

Accuracy: Accuracy refers to the ability of the data mining method to correctly detect database intrusions. The higher the accuracy, the fewer false alerts generated by the method.

Prediction Time: Prediction time refers to the time required for the data mining method to process the data and produce results. For effective intrusion detection, it is important to choose a method that is capable of processing data in real time.

Scalability: In the case of big data, it is important to choose a model that can evolve according to the growth of the data collected.

4.2.2 Evaluation of the Different Types

Classification and clustering are two popular techniques used in data mining to solve pattern recognition and data grouping problems. Although they have similar goals, they differ in terms of accuracy, prediction time and scalability.

Accuracy: When it comes to accuracy, classification is considered more accurate than clustering. This is because the classification is supervised and uses feedback to correct errors, thereby maximizing the accuracy of predictions. Clustering, on the other hand, is an unsupervised process and does not use feedback to correct errors, which can lead to lower prediction accuracy.

Prediction time: In general, classification takes longer than clustering to make predictions because it has to perform deeper analysis of the data to determine the relationships between variables. Clustering, on the other hand, is faster because it does not require such in-depth data analysis.

Scalability: Clustering is often considered more scalable than classification because it can process larger amounts of data in a short time. Classification, on the other hand, can become slower and less efficient as the amount of data increases, which can make it difficult to use for large data sets.

4.2.3 Choice and Rationale

Finally, we opt to combine the two techniques to benefit from the advantages of each of them and obtain more precise and reliable results (Table 5).

Table 5. Summary table

Criteria	Classification	Clustering
Precision	High	Moderate
Prediction time	Moderate to high	Fast
Scalability	Low to moderate	High

4.3 Choice of Data Mining Models

4.3.1 Classification

The different types of classification algorithms are:

- **Regressions**: such as logistic regression, which predicts the probability of belonging to a given class using a regression function.
- **Decision trees**: such as C4.5 and ID3, which use a tree structure to separate data into smaller subgroups and classify data based on different conditions on the features.
- **Nearest neighbors**: such as KNN, which classifies an observation according to its k nearest neighbors.
- **Bayesian classification**: such as Naïve Bayes, which uses probability theorems to predict the class of an observation.

4.3.2 Clustering

The different types of clustering algorithms are:

- K-means: It is an algorithm that partitions a data set into k groups based on their geometric proximity.
- Density-based clustering (DBSCAN): This is an algorithm that finds clusters by identifying regions of high density in the data.

4.4 Solution Design

In this part, we will design our new intrusion detection architecture and present its components.

4.4.1 Presentation of the Solution

Our solution offers a particular approach for intrusion detection in big data databases. It incorporates the use of honeypots to collect information on potentially malicious activity, combined with classification and clustering algorithms to identify malicious behavior. This method enables fast and effective detection of intrusions and dangerous activities, thus minimizing the risks to databases. The algorithms used are trained on representative data to guarantee their reliability. To implement such a solution, we have identified several key steps, namely: setting up a honeypot, collecting and preparing data, training algorithms and intrusion detection.

4.4.2 Functional Architecture

The functional architecture of our solution is as follows (Fig. 4):

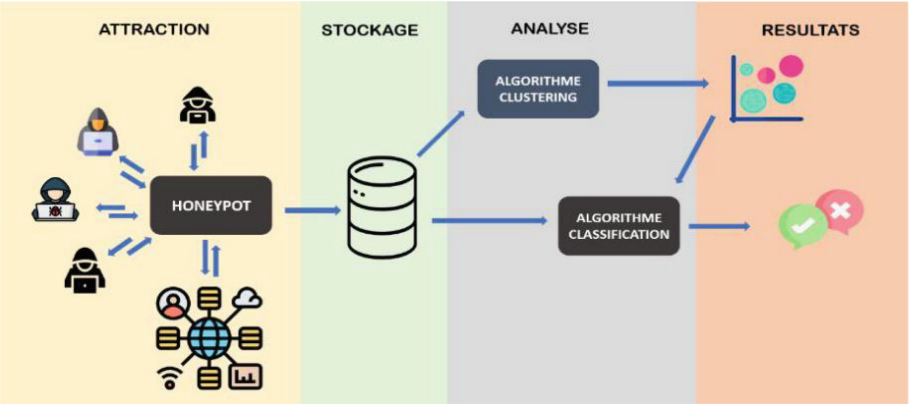

Fig. 4. Functional architecture of the solution

a) Attraction area

This entry area includes:

- Attackers such as hackers.
- A honeypot with a high level of interaction which serves as a trap for potential attackers. It collects information on attack techniques and intruder behaviors to help better understand and prevent threats.

b) Storage area
c) Analysis area

This input area includes:

- Clustering algorithm: it groups suspicious activities into different groups based on their similar characteristics using data collected by the honeypot.
- the Classification algorithm: it analyzes the data collected by the honeypot and the groups formed by the clustering algorithm to determine whether they represent a threat or not. By using the information from the clustering for a more detailed analysis, it thus reinforces the precision of its results and minimizes false positives.

d) Results area

It corresponds to the area where the results of the different analyses of our clustering and classification algorithms are stored.

5 Implementation

5.1 Presentation of the Tools

For the implementation of our solution, we made use of the tools presented in Table 6 below:

Table 6. Implementation tools

TOOLS	PRESENTATION
Python	It allowed us to write machine learning programs faster and more efficiently using Scikit-learn, Pandas, Numpy.
Google Colab	It allowed us to write and run code in Python, document our code that supports math equations, import external datasets, have free cloud services with free GPU, integrate Numpy , Pandas, Scikit-learn.
Ubuntu	It served as our host OS for installing our honeypot.

5.2 Implementation Approach

In order to solve the problem of intrusion detection in big data databases, we use classification and clustering algorithms. This choice is motivated by a study, we want to compare these different algorithms to make a better proposal for a classification method and/or technique.

5.2.1 Collecting Data with the Honeypot (Honeyd)

Data Collection is Done Using the HoneyD Tools
It allowed us to log connection attempts and malicious queries to the simulated NoSQL-like database. It also allowed us to collect information on the modus operandi of the attackers. This information helps with accuracy in predicting intrusions.

5.2.2 Clustering and Classification

The steps of our analysis are as follows:

– Importing modules and loading data

In this step we load the data as well as the different python libraries that will allow us to use the data. We have the **KDD Cup 99 datasheet** to train our model. This datasheet includes information about attacks.

5.2.3 Data Processing

This analysis step consists of understanding and visualizing the data to assess their quality and relevance for the model. It involves correcting errors and inconsistencies, transforming data to fit algorithms, and determining the most important variables to include in the model. It also helps uncover correlations between variables, hidden trends and patterns present in the data.

– **Scaling qualitative attributes**

Encoding consists of giving codes to column values that are text.

– **Reduction or selection of features**

Feature selection involves choosing the most relevant variables to include in the model. We use principal component analysis (PCA). This step reduces the dimensions of the dataset for better analysis.

5.3 Application of Clustering Algorithms

After these steps we select the parameters associated with each clustering algorithm.

5.3.1 Dbscan

- Choice of parameters

We use the grid method for the choice of parameters (distance and minimum points).

- Application of DBSCAN (Fig. 5)

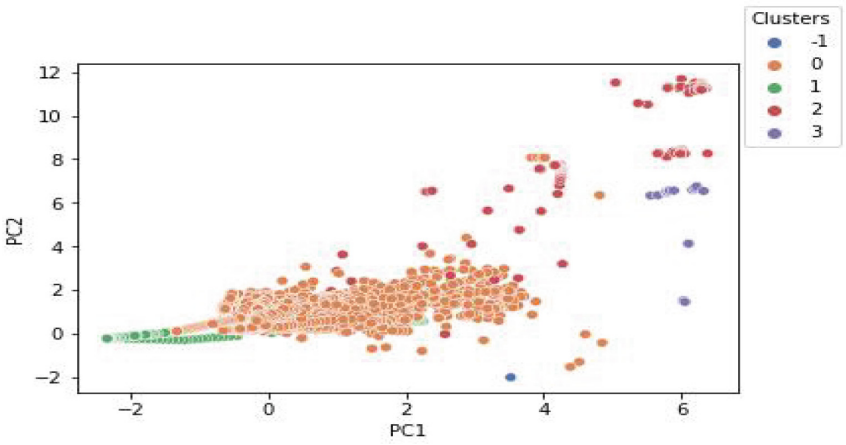

Fig. 5. dbscan algorithm results

We obtain 5 clusters, one of which consists of outliers.

5.3.2 K-Means

- Choice of parameter K (number of clusters)

The parameter K represents the number of clusters to form, we use the Elbow method to find the number of clusters to form (Fig. 6).

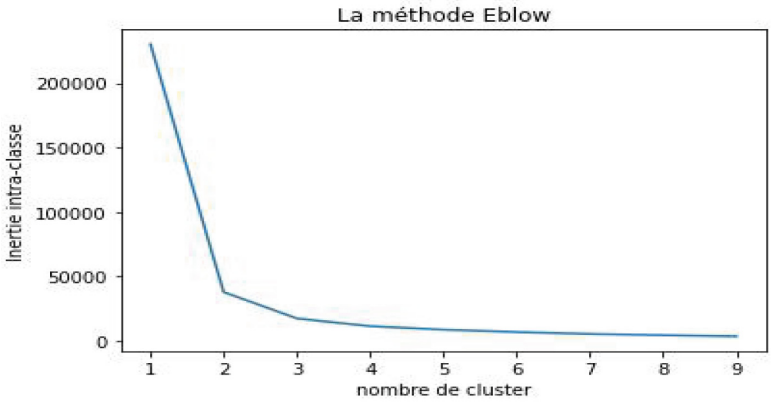

Fig. 6. Result after applying the Elbow method

The elbow of the graph represents the number of clusters, here the value of K is therefore **3.**

- Application of K-MEANS

Here is presented the different cluster shapes using K-means (Fig. 7).

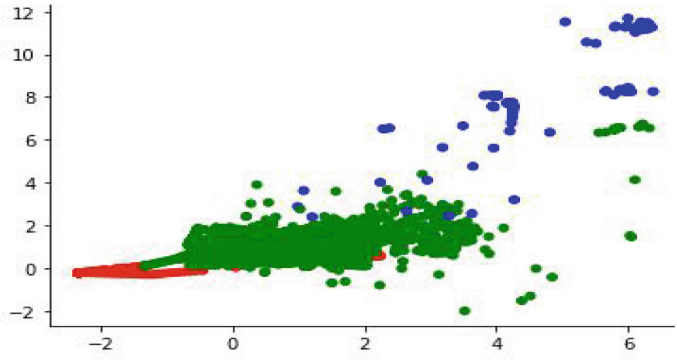

Fig. 7. Result after application of the K-means method

5.4 Application of Classification Algorithms

We use in the rest of our approach the clusters obtained after application of the clustering algorithms for the training of the classification models. And here are the different results obtained (Table 7).

Table 7. Results after classification

Methods	Regression	Naives Bayes	Decision tree	K-NN
K-means	95.49%	89.98%	99.60%	99.37%
BDBSCAN	89.57%	90.71%	97.60%	96.52%

Comments
The use of the honeypot (honeyD) associated with the different classification algorithms allows us to note that the decision trees and the KNN respectively give satisfactory results of 99.6% and 99.37% while the dbscan results are less important than the k-means.

6 Discussion

Our study aimed to propose a data mining method based on the use of honeypots in Big Data databases to improve the detection of computer attacks. We explored several data mining techniques, including classification, anomaly detection, and clustering, to extract relevant information from the data collected by honeypots.

For the application of our approach, we performed our simulations using the Honeyd tool to create an environment of vulnerabilities to attract potential attackers. We also used NoSQL databases to manage the collected data and apply data mining techniques. Our results showed that the use of honeypots (honeyD) in Big Data databases is an effective method to detect computer attacks. In particular, the clustering technique combined with the classification algorithm made it possible to identify abnormal traffic present in the database.

However, the need to have a well-designed Big Data infrastructure to manage the massive data collected by honeypots and also the need to update them regularly to remain relevant in the face of new threats could improve the security of the infrastructures.

Furthermore, our study showed that the use of honeypots in Big Data databases was a promising method to improve the detection of computer attacks by exploiting data mining techniques.

7 Conclusion and Prospects

In this work, our aim was to propose honeypot-based data mining methods for the discovery of intrusions in Big Data databases. Our major concern was how to detect different intrusions into different big data database categories using honeypots and data mining

techniques. The main objective of this study was therefore the establishment of a framework for securing data from big data. In order to achieve this objective, we first carried out a state of the art in order to present the different works from the field of intrusion detection. Then we presented the different honeypots and made a classification by functionality in order to make a choice to apply to the different types of big data database. Then we studied the different data mining techniques to highlight the different intrusions. All this allowed us to implement an intrusion detection architecture that integrates the use of the honeypot (Honeyd) to collect information on potentially malicious activities, combined with classification algorithms such as regression, decision, and clustering such as k-means, DBSCAN to identify possible intrusions into databases.

As a follow-up to this intrusion detection work, we could set up an expert system dedicated to intrusion detection based on honeypots to feed the system's knowledge base.

References

1. Mahamadou, B., Diday, E.: Data mining report Analysis of indebtedness by level of development of countries. Docplayer.fr. Consulted on 11 November 2022
2. Sellami, L.: Data Mining Approach for Intrusion Detection, p. 15. Accessed 11 November 2022
3. Big Data: Big Data Definition. https://www.lebigdata.fr/definition-big-data. Accessed 10 Dec 2022
4. https://www.universitylib.com/introduction-to-big-data/
5. Oracle: What is Big Data? Oracle.com. https://www.oracle.com/fr/big-data/what-is-big-data/. Accessed 17 Dec 2022
6. Iyad, K., Malek, S.: A dynamic honeypot design for intrusion detection. IEEE ACS/International, Washington, DC, USA, Consulté le 8 janvier 2023. ISBN: 0-7803-8577-2
7. Portokalidis, G., Bos, H.: SweetBait: Zero-hour worm detection and containment using honeypots. J. Comput. Netw. Special Issue Secur. Self-Protect. Self-Healing Syst., Consulté le 11 janvier 2023
8. David, D., et al.: HoneyStat: local worm detection using honeypots. In: Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID), pp. 39–58. Consulté le 11 février 2023
9. Zpitzner, L.: Honeypots: Tracking Hackers, Addison Wasley Professional, Septembre 2002. ISBN - 0321108957
10. Fekolkin, R.: Intrusion detection & prevention system: overview of snort & suricata. Internet Security, A7011N, Lulea University of Technology, pp. 1–4.
11. Chi, R.: Intrusion detection system based on snort. In: Liu, X., Ye, Y. (eds.) Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 3, pp. 657–664. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-40633-1_82
12. Réseaux et Sécurité Informatique: Les IDS/IPS SNORT (eventus-networks.blogspot.com). Consulté le 10 July 2023
13. Ghafir, I., Prenosil, V., Svoboda, J., Hammoudeh, M.: A survey on network security monitoring systems. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Vienna, Austria, pp. 77–82 (2016). https://doi.org/10.1109/W-FiCloud.2016.30
14. Singh, A.P., Singh, M.D.: Analysis of host-based and network-based intrusion detection system. Int. J. Comput. Netw. Inf. Secur. **6**(8), 41–47 (2014)

15. Zpitzner, L.: Honeypots: Tracking Hackers. Addison-Wesley Professional (2002). ISBN - 10: 0321108957
16. Majorczyk, F.: Detection of behavioral intrusions by diversification of COTS: application to the case of web servers" thesis, Doctorate in Computer Science, University of Rennes I, 2008, 182p. Consulted on 16 November 2022
17. Bouzayani, H.: Quantitative model for intrusion detection. An IDS-HONEYPOT collaborative architecture, Master's thesis, University of Quebec in Outaouais (UQO), 81p. (2012)
18. Diallo, A.: State of the art and prospects for a solution against DoS, final dissertation, Master in Software Engineering. Assane SECK University of Ziguinchor UFR Sciences et Technologies, 84p. (2020)
19. Rania, D.: "Network intrusion detection system based on the KNN classification algorithm" end-of-study project, Master in Information Systems Security, SAAD DAHLAB University of Blida 1, 2019, p. 63