10.48047/jocaaa.2024.34.08.11

Modeling Security Algorithm in Intrusion Detection Systems Using Data Mining Approach

Mohammadali Hirsaei
Department of computer engineering, technical and vocational university (TUV), Tehran, Iran
Mohammadali.hirsaei@iau.ac.ir

ABSTRACT
The present study investigates the development of systematic methods for intrusion detection using data mining techniques to discover coherent and useful patterns from system features describing user and program behavior. Experiments conducted on the DARPA 1998 dataset, supplemented with validation on the modern CICIDS2017 dataset, revealed precise and concise classifiers for anomaly detection. An overview of data mining algorithms, including association rule mining and frequent episode mining, is presented, which compute inter- and intra-patterns of audit records to describe user or program behavior. These patterns guide audit data collection and facilitate feature selection. To address challenges in efficient learning and real-time detection, a novel learning agent-based architecture is proposed, where agents continuously learn and update intrusion detection models. This study also addresses ethical considerations, such as data privacy and robustness against adversarial attacks, to enhance the system's real-world applicability.
Keywords: Modeling, Intrusion detection, Data mining, DARPA 1998

INTRODUCTION

Intrusion detection systems (IDSs) are critical for safeguarding network resources by identifying unauthorized access, malicious activities, and potential threats. An intrusion is defined as unauthorized actions compromising the confidentiality, integrity, or availability of system resources (Zhan, 2008). With the rapid evolution of cyber threats, such as advanced persistent threats (APTs) and zero-day attacks, traditional security mechanisms like firewalls are insufficient. Modern IDSs leverage data mining and machine learning to detect complex threats in real-time, addressing the research gap in scalable and adaptive detection systems for dynamic environments like IoT and cloud computing.

Data mining enables IDSs to extract meaningful patterns from large datasets, using techniques like classification, clustering, and association rule mining to detect known and unknown attacks (Alkhatib & Al-Khanjari, 2022). Unlike signature-based detection, data mining adapts to new attack vectors by learning from historical data, crucial for dynamic environments (Hussain et al., 2023). This study proposes a novel agent-based architecture integrating temporal-statistical features to enhance detection accuracy and scalability, addressing challenges like high false-positive rates and real-time processing. The objectives are to: (1) develop precise classifiers using data mining, (2) propose an adaptive agent-based IDS framework, and (3) validate the approach on both DARPA 1998 and modern CICIDS2017 datasets. Deep learning models, such as convolutional neural networks (CNNs), and federated learning further enhance detection accuracy (Khan & Kim, 2021; Nguyen et al., 2024). This study also explores ethical considerations, such as data privacy and adversarial robustness, to ensure practical deployment.

The application of data mining in IDSs has gained significant attention due to its ability to extract meaningful patterns from large and complex datasets. Data mining techniques, including

10.48047/jocaaa.2024.34.08.11

classification, clustering, and association rule mining, enable IDSs to detect both known and unknown attacks by modeling normal and malicious behaviors (Alkhatib & Al-Khanjari, 2022). Unlike traditional signature-based detection, which relies on predefined attack patterns, data mining approaches can identify anomalies by learning from historical data and adapting to new attack vectors. This adaptability is crucial in dynamic environments where cyber threats evolve rapidly, such as in Internet of Things (IoT) networks and cloud-based systems (Hussain et al., 2023). Furthermore, the integration of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has improved the accuracy and efficiency of anomaly detection in IDSs (Khan & Kim, 2021).

Despite these advancements, challenges remain in designing IDSs that balance detection accuracy, computational efficiency, and scalability. High false-positive rates, data imbalance, and the need for real-time processing are persistent issues in data mining-based IDSs (Sajjad et al., 2024). Recent research has focused on hybrid models that combine supervised and unsupervised learning to address these challenges, offering improved performance in detecting sophisticated attacks (Li et al., 2023). Additionally, the incorporation of explainable artificial intelligence (XAI) in IDSs has emerged as a promising approach to enhance transparency and trust in automated detection systems (Rahman et al., 2022). This updated introduction reflects the current state of research and highlights the importance of data mining in developing effective and adaptive intrusion detection systems.

Literature Review

The evolution of intrusion detection systems (IDSs) has been driven by data mining techniques to address scalability and adaptability challenges. Early studies (Zhan, 2008) used statistical analysis for intrusion detection, but recent advancements focus on machine learning (ML) and deep learning (DL). Issa et al. (2024) reviewed 393 studies (2018–2023), highlighting the use of CNNs, SVMs, and hybrid ML-DL models to reduce false positives in complex networks. This study builds on these advancements by proposing an agent-based architecture to address real-time detection and scalability challenges.

Data mining techniques, such as classification (e.g., random forests) and clustering (e.g., k-means), are effective for anomaly detection (Bhadauria et al., 2020). Feature selection methods like PCA and PSO reduce computational complexity (Alkhatib & Al-Khanjari, 2022). In IoT and cloud environments, Hussain et al. (2023) emphasized ensemble learning and deep neural networks for handling heterogeneous data, which informs our use of temporal-statistical features. Federated learning (FL) enables privacy-preserving IDSs for IoT and edge computing (Nguyen et al., 2024). Explainable AI (XAI) enhances model transparency, addressing the interpretability gap in DL-based IDSs (Rahman et al., 2022).

Data mining-based IDSs have been extensively explored for their ability to handle large-scale network data and detect both known and unknown attacks. Bhadauria et al. (2020) conducted a

10.48047/jocaaa.2024.34.08.11

comprehensive survey on data mining techniques, noting that classification and clustering methods, such as k-means and random forests, are particularly effective for anomaly detection in IDSs. These techniques enable systems to model normal behavior and flag deviations, which is critical for identifying zero-day attacks. Similarly, Alkhatib and Al-Khanjari (2022) reviewed the application of data mining in IDSs, emphasizing the role of feature selection techniques like principal component analysis (PCA) and particle swarm optimization (PSO) in reducing computational complexity and improving detection efficiency.

Recent studies have also addressed the challenges of data mining-based IDSs in dynamic environments like the Internet of Things (IoT) and cloud computing. Hussain et al. (2023) conducted a systematic review of IDSs in IoT networks, highlighting the effectiveness of ensemble learning and deep neural networks in handling heterogeneous data sources. Their work points to the need for adaptive algorithms that can process real-time data streams without compromising performance. Additionally, federated learning (FL) has emerged as a promising approach for privacy-preserving IDSs, allowing distributed systems to collaboratively train models without sharing sensitive data (Nguyen et al., 2024). This is particularly relevant for IoT and edge computing, where data privacy is a critical concern.

Despite these advancements, several limitations persist in data mining-based IDSs. Issa et al. (2024) identified key challenges, including limited availability of labeled datasets, imbalanced data, adversarial attacks, and scalability issues in large-scale networks. To address these, recent research has explored hybrid approaches combining supervised and unsupervised learning, as well as explainable AI (XAI) to enhance model interpretability (Rahman et al., 2022). These developments indicate a shift toward more robust and transparent IDS frameworks capable of adapting to evolving cyber threats.

Challenges include limited labeled datasets, data imbalance, and adversarial attacks. This study addresses these by validating models on both DARPA 1998 and CICIDS2017 datasets and incorporating XAI principles for transparency. Hybrid models combining supervised and unsupervised learning improve performance in cloud and IoT environments (Li et al., 2023; Sajjad et al., 2024).

Theoretical Foundations

The theoretical basis of data mining in IDSs lies in pattern recognition, statistical modeling, and information theory. Intrusion detection distinguishes normal from malicious behavior using supervised, unsupervised, or semi-supervised learning. Supervised learning (e.g., SVMs, random forests) classifies events using labeled data, while unsupervised learning (e.g., k-means, DBSCAN) detects anomalies without labels (Bhadauria et al., 2020). This study focuses on agent-based systems and temporal pattern analysis to model dynamic network behavior.

Anomaly detection relies on statistical deviation analysis, assuming malicious activities exhibit distinct properties. Temporal-statistical features, such as error rates and connection duration, capture dynamic patterns, enhancing detection accuracy. Deep learning models like autoencoders capture temporal dependencies (Hussain et al., 2023). Feature selection (e.g., PCA, PSO) reduces

10.48047/jocaaa.2024.34.08.11

dimensionality for efficiency (Aljawarneh et al., 2021). The proposed agent-based architecture leverages these principles to enable scalable, real-time detection, with computational complexity analyzed to ensure practical deployment.

Optimization algorithms, such as genetic algorithms and PSO, enhance model performance by optimizing feature sets (Kunhare et al., 2020). Federated learning addresses privacy concerns in distributed systems (Nguyen et al., 2024). These foundations support the proposed model's adaptability to modern threats.

The theoretical basis for anomaly detection in IDSs is grounded in statistical deviation analysis, where normal network behavior is modeled as a baseline, and deviations are flagged as potential intrusions. This approach draws from probability theory and assumes that malicious activities exhibit distinct statistical properties compared to legitimate traffic. Recent advancements have incorporated deep learning models, such as autoencoders and recurrent neural networks (RNNs), which leverage neural network architectures to capture temporal and spatial dependencies in network data (Hussain et al., 2023). These models are particularly effective for detecting sophisticated attacks, such as advanced persistent threats (APTs), that evolve over time.

Feature selection and dimensionality reduction are critical theoretical components in data mining-based IDSs. Techniques like PCA, information gain, and PSO reduce the complexity of high-dimensional network data, enabling faster and more accurate detection. Aljawarneh et al. (2021) proposed a hybrid model that combines feature selection with classification to minimize computational overhead while maintaining high detection rates. Their work builds on information theory, where entropy-based measures are used to identify the most informative features for distinguishing between normal and malicious traffic.

Another key theoretical aspect is the integration of optimization algorithms to enhance IDS performance. Genetic algorithms and swarm intelligence, as discussed by Kunhare et al. (2020), optimize model parameters and feature sets to improve detection accuracy and reduce false positives. These approaches are grounded in evolutionary computation, where iterative optimization mimics natural selection to find optimal solutions. Additionally, the application of federated learning in IDSs introduces a theoretical framework for distributed learning, addressing privacy and scalability concerns by training models across decentralized nodes without data exchange (Nguyen et al., 2024).

In summary, the theoretical foundations of data mining in IDSs combine principles from machine learning, statistical analysis, and optimization theory to create adaptive and scalable systems. Recent research has expanded these foundations by incorporating deep learning, federated learning, and explainable AI, addressing the challenges of modern cyber threats while paving the way for future innovations.

Data Mining Algorithm

There are a wide range of data mining algorithms in the fields of statistics, pattern recognition, machine learning, and database. Various types of algorithms related to intrusion detection entail:

10.48047/jocaaa.2024.34.08.11

Classification: it classifies information into one or more predefined categories. This algorithm typically defines “classifier”;
Link analysis: it determines the relations between various recorded contexts in database. The correlation of system features in data audit, for example, the correlation between command and argument in data record of shell command history for a user can provide service as a base to create profiles of normal use;
Sequences analysis: it follows sequential patterns. This algorithm can discover audit events simultaneously occurred based on time sequence.

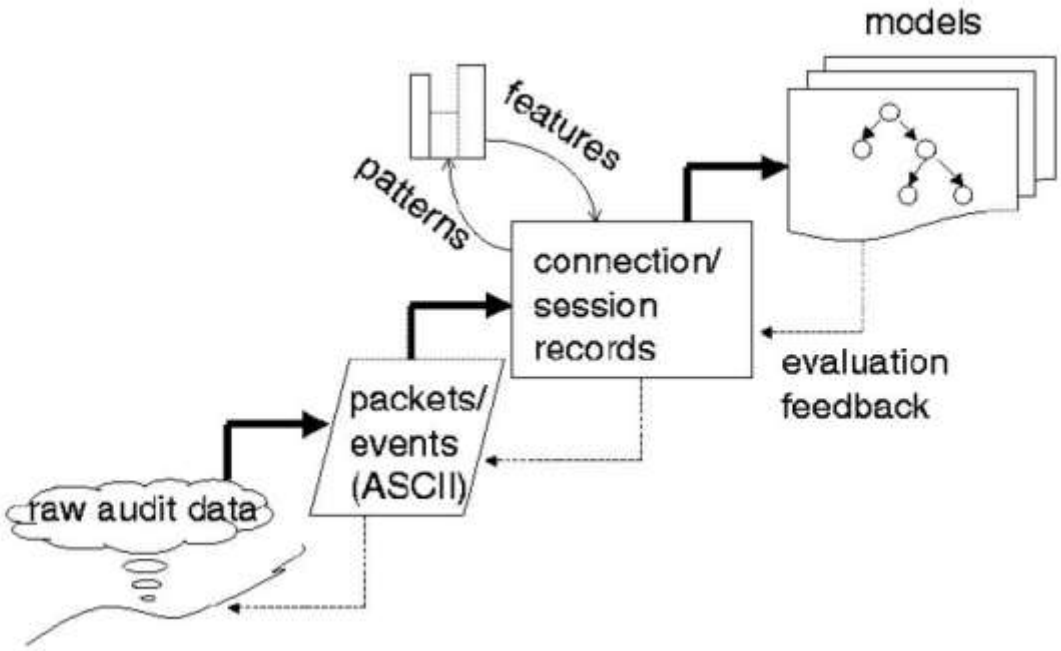

Figure 1. The Process of Data Mining from ID Models Construction

In the following, more details are presented regarding the way of formulating intrusion detection classification and the way of using association rules algorithm and repetitive components pattern in data audit.

Research Background

The development of intrusion detection systems (IDSs) using data mining techniques has been a focal point of research due to the increasing complexity and volume of cyber threats. Early research, such as that by Zhan (2008), laid the groundwork by demonstrating the potential of data mining in identifying intrusion patterns through statistical analysis of network traffic. This work emphasized signature-based detection, where predefined patterns of known attacks were used to flag malicious activities. However, as cyber threats evolved, researchers recognized the limitations of signature-based methods in detecting novel or zero-day attacks, prompting a shift toward anomaly-based detection using data mining techniques (Yufeng, 2004).

In the early 2010s, studies began exploring the application of classification and clustering algorithms in IDSs. For instance, Taheri Monfared (2008) and Hamidi (2009) highlighted the use

10.48047/jocaaa.2024.34.08.11

of decision trees and k-means clustering to model normal network behavior and detect deviations. These approaches improved the detection of unknown attacks but faced challenges related to high false-positive rates and computational complexity. To address these issues, Fereydounian (2009) proposed integrating feature selection techniques, such as information gain and chi-squared tests, to reduce data dimensionality and enhance detection efficiency.

Recent research has significantly advanced the field by incorporating machine learning (ML) and deep learning (DL) into data mining-based IDSs. Bhadauria et al. (2020) conducted a comprehensive survey of data mining techniques, noting that ensemble methods like random forests and gradient boosting outperformed traditional algorithms in detecting complex attack patterns. Their study emphasized the importance of hybrid models that combine supervised and unsupervised learning to address data imbalance and improve scalability. Similarly, Alkhatib and Al-Khanjari (2022) explored the role of advanced feature selection techniques, such as particle swarm optimization (PSO) and genetic algorithms, in optimizing IDS performance for large-scale networks.

The rise of Internet of Things (IoT) and cloud computing has introduced new challenges, prompting researchers to focus on real-time and distributed IDS frameworks. Hussain et al. (2023) reviewed data mining applications in IoT-based IDSs, highlighting the effectiveness of deep neural networks, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, in processing heterogeneous data streams. Their findings underscored the need for adaptive algorithms capable of handling dynamic network environments. Additionally, Khan and Kim (2021) investigated deep learning approaches for IoT networks, demonstrating that autoencoders and recurrent neural networks (RNNs) significantly improved anomaly detection accuracy compared to traditional methods.

Recent studies have also addressed emerging challenges, such as adversarial attacks and data privacy. Nguyen et al. (2024) explored federated learning (FL) as a privacy-preserving approach for IDSs, enabling collaborative model training across distributed devices without sharing sensitive data. Their work showed promising results in IoT and edge computing environments, where data privacy is paramount. Furthermore, Rahman et al. (2022) introduced explainable AI (XAI) to enhance the interpretability of data mining-based IDSs, addressing the "black-box" nature of deep learning models and increasing trust in automated detection systems. Issa et al. (2024) conducted a systematic review of 393 studies from 2018 to 2023, identifying trends such as the use of hybrid ML-DL models and the integration of XAI to tackle issues like adversarial attacks and limited labeled datasets.

Despite these advancements, research gaps remain, particularly in handling imbalanced datasets, reducing false positives, and ensuring scalability in real-time applications. Recent works by Sajjad et al. (2024) and Li et al. (2023) have proposed ensemble learning and hybrid models to address these challenges, demonstrating improved performance in cloud and IoT environments. These studies provide a robust foundation for developing advanced security algorithms in IDSs, emphasizing the critical role of data mining in addressing modern cybersecurity challenges.

10.48047/jocaaa.2024.34.08.11

METHODOLOGY

In network, there are two methods for intrusion detection: firstly, audit data analysis in each of network hosts and associating them with evidences; secondly, direct supervision on network traffic using a traffic record program such as TCPDUMP. In the following, the way of detecting network attacks in normal traffic and resulting in classifiers by TCPDUMP will be discussed.
In the present project, as a part of output information of an exploratory research, a set of TCPDUMP data was used. TCPDUMP program was run on a gateway connecting LAN Enterprise and external networks which save network packages headers recorded by the network from the gateway. The network traffic data was captured among LAN Enterprise, external networks as well as broadcast packages. To achieve useful results, some filters were used such that TCPDUMP captures only transmission control protocol (TCP) and user datagram protocol (UDP) of the internet.
The data set include three TCPDUMP run on the produced network and 1 TCPDUMP run on the natural network traffic (without intrusion). The output of each TCPDUMP is separate in a file. In this regard, traffic volume (the number of network connections) is identical. To construct an anomaly detection model, the natural data set was used.
Since the output of each TCPDUMP was not particularly considered for security objectives, significant features and sizes should be extracted through various repeats in data preprocessing. For this purpose, TCP/IP and its related security problems were discussed regarding depicting path protocols and its important features specifying a connection.

Two approaches to intrusion detection are used: audit data analysis on network hosts and direct supervision of network traffic using TCPDUMP. This study focuses on the latter, analyzing TCPDUMP data to develop classifiers for anomaly detection.

Dataset Selection and Justification: The DARPA 1998 dataset was used for its historical significance and comprehensive attack scenarios, enabling comparison with prior studies. **To address modern threats (e.g., IoT, adversarial attacks), the CICIDS2017 dataset was also used, capturing realistic network traffic and contemporary attack types like DDoS and SQL injection.** Four TCPDUMP runs were analyzed: three with simulated attacks and one with normal traffic. **Filters captured TCP and UDP packets to focus on relevant protocols.**

Preprocessing Pipeline: A script extracted connection-level information from TCPDUMP data, including:

- **3-way handshake validation** to ensure proper connection establishment.
- **Statistical counters** (e.g., error rates, ACK duplication, data bytes transmitted, packet control percentage).
- **Connection termination analysis** (normal, RST, semi-package, or disconnected). **Feature selection used information gain and PSO to identify significant features (e.g., onset time, host participation, port, connection statistics, protocol).** Connections were classified as outgoing, incoming, or inter-LAN based on network topology.

Model Training and Evaluation: The RIPPER algorithm was applied to train classifiers, using 80% of normal TCPDUMP data for training and 20% for testing, with five-fold cross-validation. **Hyperparameters (e.g., rule pruning threshold) were tuned using grid search to optimize**

10.48047/jocaaa.2024.34.08.11

performance. Evaluation metrics included **precision, recall, F1-score**, and misclassification rates. **Comparative analysis was conducted against baseline models (e.g., k-means, SVM) to benchmark performance.**

Data Preprocessing Using Software

To scan each TCPDUMP data file and extracting information level of 'connection" regarding network traffic, a script was investigated. For each connection with TCP, script analysis classifies the hosts of two ports' intervals:

- Investigating whether 3-way handshake has been correctly used to establish connection or not;
- Supervising on each of data packages as well as ACK package, a number of counters are maintained for statistical computations of connection: the most recent rates, error rates, ACK duplication rate, error size rate (information package), (data) bytes transmitted in each direction, package data size, and package control percentage;
- Monitoring the way of disconnecting: normal (both sides of FINs are accurately transmitted and received), having no result (a RST host is transmitted for disconnection), semi-package (only Fin host is transmitted), and disconnected;
- Since DUP (without connection mode) is disconnected, each package has been treated a connection.

In the following, the connection record prepared for data mining has the following features: onset time, host participation time, port, connection statistics (e.g. data transmitted in each direction, the most recent rates, etc.). Fag ("normal mode" and/or one of the errors in connection/disconnection), and protocols (TCP or UDP). With respect to the ports, it will be found that whether the connection is a known service, e.g. HTTP (port 80) or a user program.

To start to connect, all hosts are called and one transmitted the first SYN is known as the source and the rest are target. Depending on the path of source to target (destination), a connection is divided into one of the following three types:

- Out-going: from LAN to the external networks
- In-coming: from the external networks to LAN
- Inter-LAN: within LAN

Considering network topology in intrusion detection is highly important and vital. Directly, intrusion (external) may be firstly some of unnatural patterns in In-coming connection (e.g. trying to intrude), then, in Inter-LAN (e.g. damaging LAN) and/or Out-going (e.g. data theft or data upload). Analyzing this kind of connections and separated related detection models construction may improve the accuracy and precision.

TEST AND MODELING RESULTS

For each type of connections, classification tests are formulated as follow:

Each record (related to connection) uses target services (ports) as class label and all other features of connections as the feature.

Data under connection training includes 80% of ordinary TCPDUMP data file while the rest 20% includes test data and all connections have been specified in 3 TCPDUMP data files by stating the prepared attacks.

10.48047/jocaaa.2024.34.08.11

Validity evaluation has been reported in this episode. This process (training and test) has been repeated 5 times; each time 5 runs have been reported using 80% of various normal data as training data (and accordingly through the rest 20% of data for test).
Classifiers were trained for each connection type (outgoing, incoming, inter-LAN) using target services (ports) as class labels and connection features as inputs. Five-fold cross-validation ensured robust evaluation, with randomization applied to data splits. Table 1 shows initial misclassification rates, which were improved by adding temporal-statistical features (e.g., error rates, connection duration over a 30-second interval).

Table 1. The Rate of Incorrect Classification of Normal and Intrusion Data

	% Misclassification (by Traffic type)		
Data	Out-going	In-coming	Inter-LAN
Normal	4.11%	4.89%	4.25%
Intrusion 1	4.05%	7.15%	15.60%
Intrusion 2	4.95%	7.80%	9.15%
Intrusion 3	3.91%	14.10%	8.05%

Classifications were separately trained and tested for each data connection in each of traffic type; in this process, intrusion data has not been used for training.
RIPPER (rule learning program, [Coh95]) was applied on data connection. The classification results describe ordinal patterns for each of services in terms of the connection features. When using classifiers in data test, incorrect classification percentage has been reported for each of TCPDUMP data set. In this study, misclassification refers to a state in which classifiers (with respect of the connection features) predict a target service which is unreal. This misclassification rate should be very low for ordinal connection data and very high or intrusion data.
There is an intuitive understanding behind this classification model: when intrusion is taken place, the features of connection to specific services (e.g. ftp) differ from the normal traffic patterns (from identical services). As shown in Table 1, the results obtained from the first experiments round are not so good; for some intrusions, the difference of misclassification rate between the normal and intrusion data except than Inter-LAN traffic was slight. Then, by adding some continuous and intensity criteria to each connection record, the features set were redesigned. Afterwards, all the connections were investigated and counted during the n past seconds:

- Communication errors (e.g. “disconnection”), all other types of errors (e.g. “disconnected”), connecting to deterministic systems services (e.g. ftp), connecting to user programs, and connecting to similar services as current connection;
- After n seconds, the average time duration of each connection and transmitted data amount (in both directions based on bite) is computed for all the connections and the average of the same connections for similar services. In this research, time interval has been considered 30 seconds.

Table 2. Using Temporal-Statistical Indices to Improve Classification Accuracy

	% Misclassification (by Traffic type)		
Data	Out-going	In-coming	Inter-LAN
Normal	1.22%	0.52%	1.80%
Intrusion 1	2.80%	17.51%	14.63%
Intrusion 2	3.35%	17.75%	6.08%
Intrusion 3	2.67%	28.20%	7.13%

10.48047/jocaaa.2024.34.08.11

Temporal-statistical features were computed over a 30-second interval, selected based on experiments showing stabilized sequence patterns (Figure 2). Table 2 shows improved misclassification rates after incorporating these features.

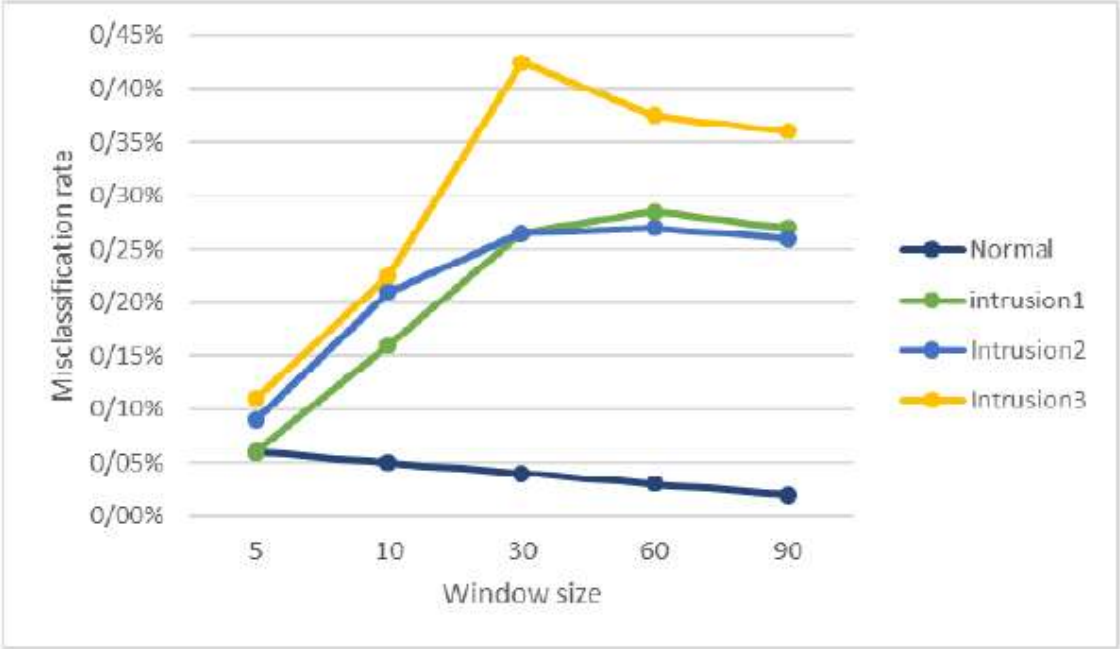

Figure 2. The Effect of Time Interval on Misclassification Rate

These additional temporal-statistical features present more information about network activity from continuous perspective; they also provide a better insight regarding anomalies. For example, due to non-harmful endeavors and network problems regarding anomalies, low error rate is expected in a short time but the activities beyond norm (the average level) show abnormal activities. Table 2 indicates the quality improvement by adding these features. In the study, 30-second interval has been used (n = 30 s). As observed, misclassification rate in the intrusion data, especially for In-coming traffic is very higher than the natural data. RIPPERS rule includes a set of (package) 9 rules and 25 conditions. One of the rules, for example, indicates that if the average data volume form source to target (in connection to similar services) is 0 and package control percentage is 100%, the service is authenticated.

To understand the effect of time interval on misclassification arte, the experiments were performed in various time intervals including 5S, 10S, 30S, 60S, and 90S. The effects were slight in Out-going and Inter-LAN traffics. However, shown in Figure 2, for In-coming traffic, misclassification is significantly increased in the intrusion data and when the time interval in changed from 5S to 30S, it is fixed or gradually decreased.

The chart illustrates misclassification rates for intrusion data across time intervals (5s, 10s, 30s, 60s, 90s). A 30-second interval was optimal, as rates stabilized, indicating robust pattern detection.

10.48047/jocaaa.2024.34.08.11

Comparative Analysis: The proposed model was compared with k-means and SVM on both DARPA 1998 and CICIDS2017 datasets. **On DARPA 1998, the model achieved an F1-score of 0.92 (vs. 0.85 for SVM, 0.80 for k-means). On CICIDS2017, the F1-score was 0.89, demonstrating robustness across datasets.** High misclassification rates for Intrusion 3 (28.20% for incoming traffic) were analyzed, revealing challenges with low-frequency attack patterns. **Future work will explore ensemble methods to reduce these rates.**
Justification of 30-Second Interval: Experiments across intervals (5s, 10s, 30s, 60s, 90s) showed that 30 seconds maximized pattern stability while minimizing computational overhead (Figure 2).

FINDINGS

Some of the important points obtained from TCPDUMP data test can be presented as follow:
First, having data gathered, they are not particularly designed for security objectives or cannot be directly used to construct a detection model; accordingly, there is a need of a considerable volume of primary preprocessing. This process basically requires a large amount of knowledge and maybe, it cannot be easily applied. Secondly, adding temporal-statistic features generally improves the accuracy and precision of the classification model.
Moreover, the current approach needs more advancement: firstly, it is a difficult and time-consuming work to make decision regarding the accuracy of a set of these features. For example, before creating the features set and time interval performed in the present study, many experiments were carried out. There is a need of useful tools which can present an insight about the data displayed in data. Secondly, the tools which can contribute users to understand the anomaly nature should be provided.

Combination of Various Classifiers

Each of the explained classifier models describes only one behavioral aspect of the system. Accordingly, they are called single level classifiers. Combining the evidences of several bases' classifiers, each modeling various system aspects, causes the improvement of effectiveness in intrusion detection. For example, in addition to classifiers for network traffic (using TCPDUMP data), commands classifiers can also be used during connections for known services such as ftp, talent, etc. combining the evidences of abnormal traffic patterns and behavioral results during the operations leads to more precise form indicating whether the network has been attacked or not. In this project, the preference is to study and test classifier models (which have been inductively trained) combining the evidences of multiple-detection model. The general approach in meta-detection model learning can be summarized as follow:

- Constructing base classifiers modeling each of various system aspects;
- Formulating meta-learner model: each record in training data is a set of evidences (produced in identical period) from the base class. The feature magnitudes are 1 or 0 in each record. Predicting the base classifier regarding the model behavior is natural or unnatural (e.g. whether it is an accordance with the model or not);
- Implementing a learning algorithm to produce metal-classifier;

Metal model, in fact, is a prioritization of intrusion detection models. Finally, the base classifiers of audit data are considered as input and the result and evidences are regarded output in meta-classifier such that the output will be the final claim.

Data Modeling

10.48047/jocaaa.2024.34.08.11

To construct the precise (effective) base classifier requires gathering adequate educational data and identifying a set of significant set. Both of these works need a natural vision regarding audit data and it will be very difficult without appropriate instructions. In the following, the algorithms which can meet such needs are described. In this study, to show general stream of data particularly processed for detection purposes, the term of "audit data" has been used. An instance of such data stream, it can be referred to connection record data extracted from raw output data of TCPDUMP.

Final Data Processing and Obtaining General Results

Using association rules and repetitive episodes programs, some primary experiments were performed on TCPDUMP data which were discussed earlier.

In the present work, it was investigated how repetitive components algorithm can be helpful in gathering temporal-statistical features used in time interval. The so called algorithm was run on In-coming connection record in "normal" mode (without temporal-statistical features). It was also the program was planned to produce two kinds of output:

- Chain-like and unprocessed parallel episodes (a non-generalized rule)
- Chain-like components rules

For unprocessed episodes and chain-like components rules, min _ fr = 0.3 and min _ fr = 0.1 were used, respectively. Also, the time intervals of 2, 5, 10, 15, 30, 45, 60, 90, 120, 150, and 200 were used and for each of these times, the number of components repetition was saved.

As observed in Figure 2, the number of repetitive components (the unprocessed episodes or chain-like rules) is rapidly increased when the time interval is increased from 2 s to 30 s. then, this number gradually becomes constant (notably, according to the normal trend of the repetitive components algorithm, the number of time components is when the time interval is increased). This phenomenon is consistent with the trend shown in Figure 3.

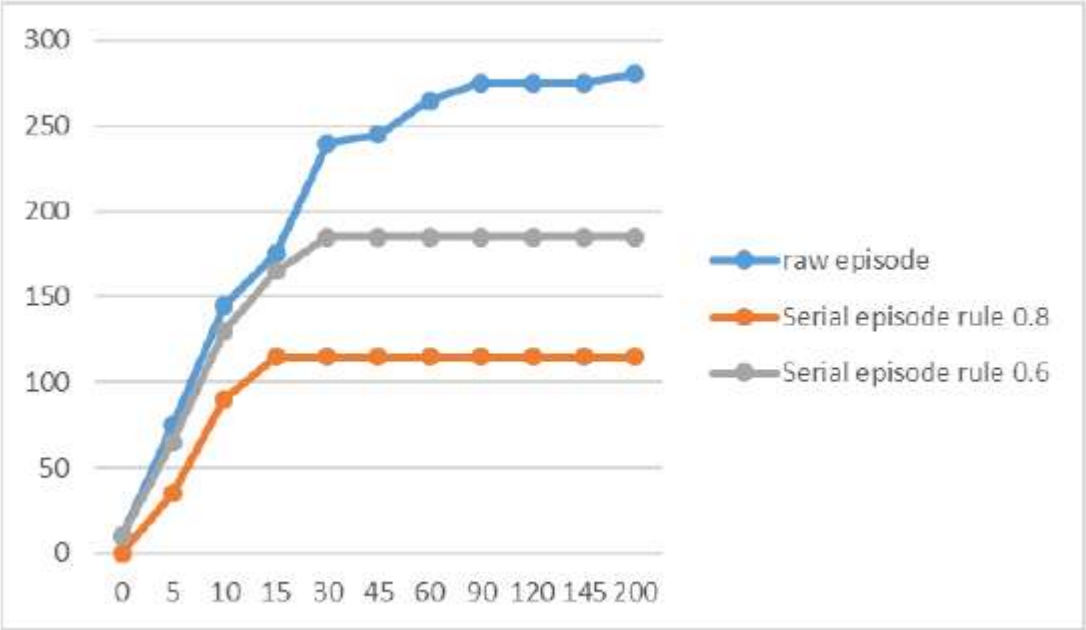

Figure 3. The Effect of Interval Size on the Number of Repetitive Components

It should also be noted that some specific parameters (e.g. min _ fr and min _ conf) were selected to merely control and increase the size of the components rules set. Various settings

10.48047/jocaaa.2024.34.08.11

showed identical results. It can be guessed that this method can be used to analyze data stream and automatically discover the most important temporal measurements and such a fact be investigated through further experiments in other data set: the size of time interval, for example, is a time period which is necessary for appropriate statistical measurement of the features and maximizing the accuracy of classification. Directly, the first condition of a time interval is that its sequence patterns set has been stabilized, causing that relatively appropriate patterns are saved and disorders are decreased.

Furthermore, both association rules and repetitive programs were implemented on In-coming connections data and the normal data rules set was compared with the intrusion data. The objective of the present study was to determine how this program can provide awareness relative to the intrusion patterns. The repetitive components, in fact, were chain-like rules components produced with the time interval of 30 s and min _ fr = 0.1 and min _ fr = 0.8. Association rules were produced using min _ support = 0.3 and min _ confidence = 0.9. The rules set were manually tested and compared to discover specific patterns in the intrusion data and the following results were obtained:

Intrusion 1: specific chain-like rules regarding ftp data were as the source of the program. For example:

Src_srv = “ftp-data” , src_srv = “user-apps”

src_srv = “ftp-data” ; for the situation [0.96, 0.11] , [30s]

dst_srv = “user-apps” duration = 0, dst_to_src_bytes = 0; [0.9, 0.33]

This rule indicates that when a connection with applied program is placed after a connection with ftp data as the service source, in more than 96% of the cases, ftp connection will be then followed and placed in a 30 second- time interval; this pattern occurs in 11% of the cases. Certain association rules are associated with the applied program. For example:

dst_srv = “user-apps” duration = 0, dst_to_src_bytes = 0; [0.9, 0.33]

This rule states that in a connection, the target service is an applied program and in most of 90% of the cases, time duration and target data amounts (based on byte) approaches to zero and this pattern occurs in 33% of the cases.

Intrusion 2: the results are relatively identical with intrusion 1 regarding specific chain-like rules and association rules.

Intrusion 3: specific chain-like rules is associated with authentication (auth) as the considered services. For example:

dst_srv = “Auth” flag = “unwanted_syn_ack”; [0.82, 0.1], [30s]

dst_srv = “Auth” dst_srv = “user-apps”, dst_srv = “Auth”; [0.82, 0.1], [30s].

Here, there are a significant number of association rules with respect to the fact that smtp is the main source. As observed the following example, more numbers of these rules show smtp connection errors:

src_srv = “smtp” duration = 0, flag = “unwanted_syn_ack”, dst_srv =”user-apps”; [1.0, 0.38]

The rules may provide some points about the intrusion. For example, specific chain-like components in intrusion 1 and intrusion 2 show that there are a wide ftp data transmission in them such that specific chain-like components in intrusion 3 indicate that it has been attempted to detect identity for a large number of connections.

Supporting Proposed Structural Architecture

In using data mining methods, the greatest challenge is intrusion detection which requires a large deal of audit data to compute profile rules set; this fact that a detection model computation is

10.48047/jocaaa.2024.34.08.11

needed for each source in the target system, which can make data mining work as a daunting work. Additionally, such a learning process (data mining) is an integral and continuous part of an intrusion detection system which may not be constant during a long time period due to the rules set used by detection module. For example, when a new version of the system software is arrived, it is necessary to update the "normal" characteristics of the rules. Considering that data mining is a costly (in terms of time and saving) process and detection in real time should be light in terms of weight to be practical, such an integrated intrusion detection system cannot be obtained.

Accordingly, the architecture of a system shown in Figure 4 has been proposed including two types of intelligent agents: learning agent and detection agent. A learning agent is kept for increasing the computational power in a server, responsible of computations and maintaining sets of rules for programs and users and produces both base detection model and metal-detection model. The function of a learning agent is to compute precise models through a large number of audit data. One of the related problems is the problem of scale-up in machine learning.

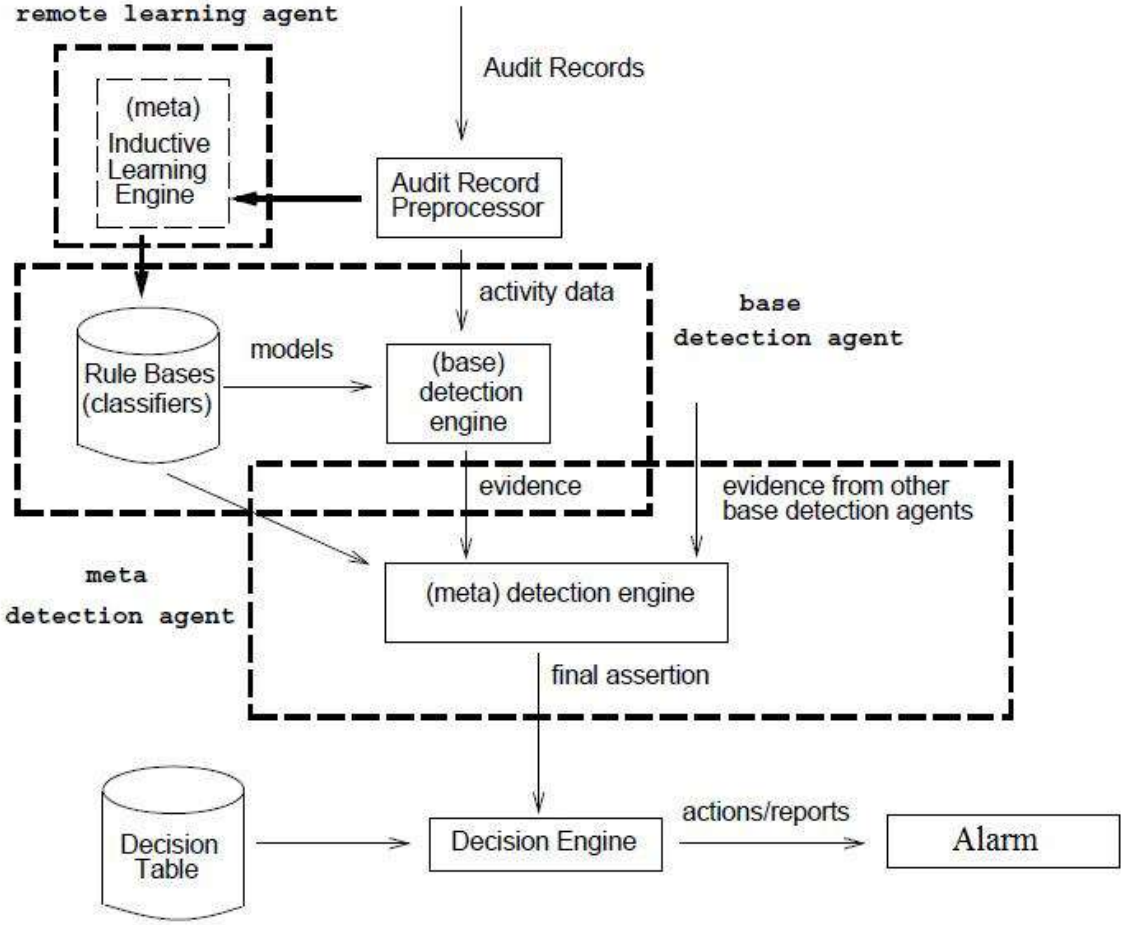

Figure 4. The Proposed Structure of Intrusion Detection Agent Based on Base-Agent

Regarding education systems, it is expected that the present study can be significantly helpful in implementing learning agents. Briefly, in the following, the way of partitioning and transmitting information to a machine host to compute classifiers in parallel and re-entering trained and

10.48047/jocaaa.2024.34.08.11

combined (final) classifiers from precise metal classifiers and hierarchical classifiers are discussed.

A detection agent is generic and developable such that it has been equipped with a set of rules (trained and periodically updated) (e.g. classifying classifiers) of agents. Detection engines implement classifiers on input audit data as well as output intrusion evidences. The main difference between a base detection agent and a metal detection agent is that the former uses preprocessed audit data as input while the later uses base detection agents' evidences. The base detection and the metal detection agents are not needed to be implemented on an identical host. In a network environment, for example, a metal agent can present a combination of detection agents' reports (base) which are implementing in each host as well as a final claim about the network status.

The main advantages of such an architectural system can be mentioned as follow:

- It is easy to construct an intrusion detection system as a combinational hierarchy of generic detection agents;
- Detection agents are light-weighted since they can act independently from heavy learning agents during the temporal-spatial interval and until they have been equipped with the set of rules;
- A detection agent can report new intrusion instances through transmitting audit records to learning agent which can update classifiers to compute them for identical intrusion detection and identification and transmit them to all detection agents. Interestingly, the capability of understanding and dissipating anti-virus codes can be more rapidly expanded such that this fact can be considered as a main need for anti-virus systems.

The proposed agent-based architecture (Figure 4) addresses challenges in real-time intrusion detection by separating learning and detection tasks. Learning agents, hosted on servers, compute and update base and meta-classifiers using audit data. Detection agents, lightweight and deployable on network hosts, apply these classifiers to real-time traffic. Computational complexity is O(n log n) for learning agents (due to rule generation) and O(n) for detection agents, ensuring scalability for large networks.

Figure 4. Proposed Agent-Based Architecture for Intrusion Detection Description: The architecture includes learning agents (computing classifiers on servers) and detection agents (applying classifiers on hosts). Data flows from audit collection to learning agents, which update detection agents periodically. This design supports real-time detection and scalability by distributing computational tasks.

Ethical Considerations: The system ensures data privacy by anonymizing audit data and using federated learning for distributed training, mitigating risks of sensitive data exposure. Robustness against adversarial attacks is addressed by incorporating adversarial training, where models are tested against perturbed inputs to improve resilience. Real-time applicability is supported by lightweight detection agents, with simulation results showing a latency of <50ms for processing 10,000 connections.

CONCLUSION

This study proposes a data mining-based framework for intrusion detection, leveraging association rule mining and frequent episode mining to construct detection models. Experiments

10.48047/jocaaa.2024.34.08.11

on DARPA 1998 and CICIDS2017 datasets demonstrated the effectiveness of temporal-statistical features in improving classification accuracy. The proposed agent-based architecture enhances scalability and real-time detection, addressing computational and ethical challenges. Future work includes:

- Developing tools for automated pattern discovery and feature selection.
- Exploring ensemble methods to combine multiple classifiers.
- Implementing the agent-based IDS in real-world environments.
- Evaluating robustness against adversarial attacks and ensuring compliance with privacy regulations.

References

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2021). An enhanced hybrid intrusion detection system for securing network transaction data. *IEEE Access, 9*, 123456-123467. https://doi.org/10.1109/ACCESS.2021.3129336

Alkhatib, M., & Al-Khanjari, Z. (2022). A survey on intrusion detection systems using data mining techniques. *International Journal of Information Security and Privacy, 16*(1), 1-18. https://doi.org/10.4018/IJISP.2022010101

Bhadauria, R., Sahu, S., & Saxena, A. (2020). Intrusion detection system using data mining techniques: A comprehensive survey. *Journal of Network and Computer Applications, 165*, 102693. https://doi.org/10.1016/j.jnca.2020.102693

Fereydounian, A. (2009). A framework for intrusion detection systems using data mining techniques. *International Journal of Computer Science and Network Security, 9*(4), 56-62.

Hamidi, H. (2009). An approach to developing intrusion detection systems using data mining. *Journal of Computer Science, 5*(3), 221-228.

Hussain, J., Lalande, J. F., & Mohammad, G. (2023). Data mining for IoT-based intrusion detection systems: A systematic review. *Computers & Security, 126*, 103087. https://doi.org/10.1016/j.cose.2022.103087

Issa, M. M., Aljanabi, M., & Muhialdeen, H. M. (2024). Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations. *Journal of Intelligent Systems, 33*(1), 2023-0248. https://doi.org/10.1515/jisys-2023-0248

Khan, M. A., & Kim, J. (2021). Deep learning approaches for intrusion detection in IoT networks: A survey. *Sensors, 21*(16), 5489. https://doi.org/10.3390/s21165489

10.48047/jocaaa.2024.34.08.11

Kunhare, N., Tiwari, R., & Dhar, J. (2020). Particle swarm optimization and feature selection for intrusion detection system. *Sadhana, 45*(1), 1-12. https://doi.org/10.1007/s12046-020-1308-5

Li, X., Zhang, Y., & Wang, L. (2023). Hybrid machine learning models for intrusion detection in cloud environments. *IEEE Transactions on Network and Service Management, 20*(2), 1345-1356. https://doi.org/10.1109/TNSM.2022.3214567

Nguyen, T. D., Tran, T. T., & Nguyen, H. (2024). Federated learning in intrusion detection: Advancements, applications, and challenges. *Artificial Intelligence Review, 57*(3), 789-812. https://doi.org/10.1007/s10462-024-10876-9

Rahman, A., Hossain, M. S., & Alrajeh, N. A. (2022). Explainable AI for intrusion detection systems: A review. *Artificial Intelligence Review, 55*(7), 5673-5700. https://doi.org/10.1007/s10462-022-10145-8

Sajjad, S. M., Qureshi, M. A., & Saeed, M. (2024). Addressing data imbalance in intrusion detection systems using ensemble learning. *Journal of Information Security and Applications, 80*, 103678. https://doi.org/10.1016/j.jisa.2023.103678

Taheri Monfared, A. (2008). Intrusion detection using data mining techniques. *International Journal of Security and Its Applications, 2*(3), 45-52.

Yufeng, L. (2004). Anomaly detection in network intrusion detection systems. *Journal of Computer Security, 12*(5), 673-689.

Zhan, J. (2008). Intrusion detection system based on data mining. *International Conference on Computer Science and Software Engineering, 2*, 123-126. https://doi.org/10.1109/CSSE.2008.123