ZW-IDS: Zero-Watermarking-based network Intrusion Detection System using data provenance

Omair Faraj
Internet Interdisciplinary Institute,
Universitat Oberta de Catalunya
CYBERCAT-Center for Cybersecurity
Research of Catalonia
Barcelona, Spain
SAMOVAR, Telecom SudParis,
Institut Polytechnique de Paris
Palaiseau, France
ofaraj@uoc.edu

David Megías
Internet Interdisciplinary Institute
(IN3),
Universitat Oberta de Catalunya
(UOC)
CYBERCAT-Center for Cybersecurity
Research of Catalonia
Barcelona, Spain
dmegias@uoc.edu

Joaquin Garcia-Alfaro
SAMOVAR, Telecom SudParis,
Institut Polytechnique de Paris
Palaiseau, France
joaquin.garcia_alfaro@telecom-sudparis.eu

ABSTRACT

In the rapidly evolving digital world, network security is a critical concern. Traditional security measures often fail to detect unknown attacks, making anomaly-based Network Intrusion Detection Systems (NIDS) using Machine Learning (ML) vital. However, these systems face challenges such as computational complexity and misclassification errors. This paper presents ZW-IDS, an innovative approach to enhance anomaly-based NIDS performance. We propose a two-layer classification NIDS integrating zero-watermarking with data provenance and ML. The first layer uses Support Vector Machines (SVM) with ensemble learning model for feature selection. The second layer generates unique zero-watermarks for each data packet using data provenance information. This approach aims to reduce false alarms, improve computational efficiency, and boost NIDS classification performance. We evaluate ZW-IDS using the CICIDS2017 dataset and compare its performance with other multi-method ML and Deep Learning (DL) solutions.

KEYWORDS

Intrusion Detection System, Data Provenance, Data Hiding, Zero-Watermarking, Machine Learning, Support Vector Machine

ACM Reference Format:
Omair Faraj, David Megías, and Joaquin Garcia-Alfaro. 2024. ZW-IDS: Zero-Watermarking-based network Intrusion Detection System using data provenance. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30–August 02, 2024, Vienna, Austria.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3664476.3670933

1 INTRODUCTION

As the digital age advances, the importance of network security has become a major issue in the cybersecurity community. The rise of information and network technologies has led to an accumulation of a huge amount of data related to organizational operations and individual activities [23, 32]. If compromised, this data could lead to significant losses and security breaches. The increase reliance on network infrastructure introduce the importance of securing these networks against malicious intrusions and cyber threats [19, 34]. To protect networks from intrusions and attacks, various approaches have been proposed and implemented, including firewalls, digital signatures, and Intrusion Detection Systems (IDS).

IDS play an important role in detecting different types of attacks, serving as a valuable tool for the in-depth defense of computer networks. They monitor network traffic for known or potential malicious activities and trigger an alarm when malicious activity is detected. IDS are generally categorized into two types: misuse and anomaly intrusion detection systems. Misuse-based IDS identify intrusions based on system weaknesses and known attack signatures, but they fail to recognize new or unfamiliar attacks. In contrast, anomaly-based IDS are based on normal behavior parameters and use them to identify any action that significantly deviates from normal behavior [9, 15, 17]. In our work, we focus on anomaly-based intrusion detection.

Machine Learning (ML) has been widely applied to anomaly-based intrusion detection. ML-based IDS provide a learning-based system to discover classes of attacks based on learned normal and attack behavior. The goal is to generate a general representation of known attacks. Misuse detection techniques fail to detect unknown attacks, although they provide good detection accuracy for detecting well-known attacks. Various ML techniques have been explored and implemented to build an anomaly-based IDS [5, 14, 24, 28]. The most widely known method in IDS is supervised learning, which builds a mapping function based on pre-defined input-output pairings. Additionally, unsupervised learning is employed, which lets a model to infer internal relationships on its own without the need for a labeled data set. Furthermore, hybrid methods are another strategy that has gained popularity in the IDS research community. In order to fully exploit the advantages of each learning technique and enhance the overall detection rate, these methods combine two or more ML techniques. They are also a useful alternative technique for reducing the bias caused by an imbalanced data set toward more frequent attacks. However, this approach increases the complexity and computational time of the learning model [2, 21].

In recent years, one of the most known supervised learning techniques, Support Vector Machine (SVM), have been successfully applied to handle complex patterns, which are nonlinear and high dimensional. SVM has proved to perform better than traditional learning approaches in terms of classification and detection of attacks in a binary and multi-class classification scenarios in network security applications. SVM provide several advantages over other ML techniques, like Deep Learning (DL). SVM provide interpretable decision boundaries, making it easier to understand and trust the model's predictions. They perform well with network traffic datasets, have faster training times, and are robust to noisy data. Additionally, SVM offer feature importance scores, help avoid overfitting, and are more resource-efficient compared to DL models, making them an effective solution for anomaly-based IDS [20, 22, 30, 33, 38]. However, IDS often deal with large volumes of data, which may contain irrelevant and redundant features. This can slow down the training and testing process, consume more resources, and result in a poor detection rate [12]. Moreover, the misclassification of attack packets as normal is still a real concern in ML-based IDS. This is called Type 1 error where the system fails to detect an intrusion or malicious activity that is actually present in the data. In other words, it is a false positive, where the IDS mistakenly classifies an instance as non-malicious when it is, in fact, malicious. This error can result in security breaches and vulnerabilities going unnoticed, posing significant risks to the system's integrity and safety. To address this issue, ML approaches need to be assisted with other security techniques to minimize the number of misclassified packets and increase detection rate. In this paper, we introduce zero-watermarking and data provenance to improve IDS performance and solve the computational complexity of other solutions that combine several ML methods.

More precisely, we present ZW-IDS, a novel approach which integrates an anomaly-based Network Intrusion Detection System (NIDS) with a zero-watermarking-based approach for data provenance. Data provenance provides the capability to ensure data trustworthiness by summarizing the history of ownership and actions performed on collected data from the source device to the final destination. While previous studies focused on modeling, collecting, and querying provenance, IDS have been overlooked. The main goal of this work is to minimize the false alarm rate, improve the computational complexity and enhance the classification performance of NIDS. Firstly, we introduce a first layer of classification using SVM by adopting a feature selection method based on Extremely Randomized Trees. Secondly, we propose a novel zero-watermarking approach using data provenance as a second layer of classification, where we use provenance information as extracted features to generate a zero-watermark for each captured data packet. Moreover, we apply these two layers of classification to build an effective anomaly-based NIDS and evaluate the effectiveness and feasibility of our approach by conducting experiments on the CICIDS2017 [31] intrusion detection dataset.

Considering the above, the main contributions of our work are summarized as follows:

- Propose a novel approach for integrating a zero-watermarking-based data provenance approach with an anomaly-based NIDS.
- Improve the classification performance of anomaly-based NIDS and reduce the high false alarm rates in intrusion detection by introducing a two-layer classification system based on SVM and zero-watermarking (data provenance).
- Evaluate the performance of the proposed approach through a set of performance metrics including accuracy, precision, recall, F-score, false alarm rate, and computational overhead, using the CICIDS2017 dataset.
- Provide a comparative analysis with existing ML and DL-based IDS.

The rest of the paper is organized as follows. Section 2 surveys previous work on ML and DL-based IDS. Section 3 presents an overview on NIDS, data provenance and zero-watermarking. The proposed model is presented in Section 4. Section 5 presents the simulation and results of the proposed model, including the analysis about the results. Section 6 closes the paper with conclusions and further research directions.

2 RELATED WORK

To the best of our knowledge, we are presenting the first integration of a data provenance approach with anomaly-based NIDS in network security, using zero-watermarking as a technique to represent provenance records, and ML classification.

There have been previous works implementing IDS in networks using different ML and DL techniques, as we discuss in this section. With the advancement of computer networks, securing its infrastructure has made intrusion detection a very important issue to implement. Various ML methods, such as Fuzzy Logic, Decision Trees (DT), K-Nearest Neighbors (KNN), SVM, Random Forest, Naive Bayes, Logistic Regression and Artificial Neural Networks (ANN) approaches, are used in IDS to distinguish between normal network activity and malicious intrusions. SVM, in particular, has shown better performance compared to standard classification methods, allowing several researchers to propose several SVM-based IDS solutions. Despite the advantages of SVM-based IDS in terms of detection accuracy and learning speed over traditional algorithms, the issue of misclassification of attack packets still needs improvement. Furthermore, a number of different techniques have been proposed in the literature aimed at enhancing traditional ML algorithms.

In a study by Tao et al. [33], a new IDS is introduced to improve detection rate, false positive and false negative rates. This system, called FWP-SVM-GA, uses a Genetic Algorithm (GA) to improve the performance of an SVM algorithm. The GA first selects the most relevant features from the data. Then, SVM parameters are optimized to achieve the highest accuracy. After training the model, the FWP-SVM-GA can effectively identify and categorize unusual network activity. Focusing on a single attribute, packet arrival rate, instead of the complex features often found in online datasets, Jan et al. [11] propose an SVM-based classifier for a lightweight IDS evaluated using CICIDS2017 network traffic dataset and a generated dataset using MATLAB™ according to Poisson distribution. The classifier's performance using linear, polynomial, and radial-basis kernel functions is analyzed and compared to other ML techniques like neural networks, KNN, and DT.

Ravi et al. [27] propose a thorough method for network intrusion detection, focusing on merging features from hidden layers of recurrent models. They explore traditional ML algorithms like Naive Bayes, Logistic Regression, KNN, DT, and Random Forest, as well as recurrent DL models –such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRUs)–. They also investigate reducing the complexity using Kernel Principal Component Analysis (KPCA) and assess performance across various intrusion datasets, leading to ensemble meta-classifiers and feature combination for improved detection.

A DL model based on LSTM for multiclass attack detection classification is proposed by Rao and Suresh Babu [26]. Enhancing the classifier's performance involves hybridizing a convolutional neural network called LeNet 5 and LSTM, and implementing Imbalanced Generative Adversarial Network (IGAN)-based class imbalance. This process can prevent the unnecessary time and space losses involved with oversampling as well as the loss of important samples due to random undersampling.

Alghushairy et al. [1] propose a Network Outlier Detection System (NODS) for classifying normal and attack network traffic. The system uses SVM and Gaussian Naive Bayes (GNB) classification algorithms to categorize the behavior of incoming network connections impacting a computer network. Both algorithms were built and assessed using network traffic datasets. Data mining preprocessing steps for network flow data, alongside optimizing Radial Basis Function (RBF) control parameters and the GNB smoothing parameter, prove to enhance the overall effectiveness of the proposed NODS.

While many techniques that use SVM or other ML and DL approaches for intrusion detection have been proposed lately, there are still some drawbacks with these approaches, such as: (1) Some algorithms might struggle with complex attacks that deviate from established patterns. Novel attacks or zero-day exploits, for instance, might bypass the detection capabilities of these models; (2) the effectiveness of ML models heavily relies on the training data (e.g., if the training data is limited or does not encompass a wide range of attack types, the model might not be able to generalize well to unseen attacks and, later on, this can lead to false positives when the system encounters attacks not included in its training set); and (3) some methods might establish static thresholds for identifying anomalies. While our third observation does not exclude detection of known attack patterns, attackers can adapt their methods to remain below these thresholds. This can trick an IDS to misclassify attacks as legitimate traffic. To handle the problem, we propose an augmented two-layer IDS approach using SVM for the first layer of classification and a zero-watermarking approach using provenance information as a second layer of detection. Even if the IDS misclassifies an attack in the first process, the other layer might still be able to prevent it from compromising the system.

3 BACKGROUND

In this section, we provide a more thorough background on the techniques we used in our approach by introducing a brief overview of NIDS, data provenance, and zero-watermarking.

3.1 Overview of Network Intrusion Detection Systems

An IDS acts like a network security guard. It constantly monitors and scans traffic for any unusual behavior that deviates from normal network activity and warns network administrators if it detect any suspicious network behavior [36]. IDS uses different techniques for intrusion detection such as signature-based or anomaly-based detection techniques [39]. IDS can be deployed in three main placement strategies as NIDS, Host-based Intrusion Detection System (HIDS), and Collaborative Intrusion Detection System (CIDS) which combines both NIDS and HIDS. An NIDS, in communication networks, is a security tool designed to monitor and analyze network traffic to detect malicious activities or unauthorized access. An NIDS enhances the security of network devices by continuously monitoring and analyzing network communications, helping to detect and mitigate potential security threats to ensure the integrity and functionality of computer networks. An NIDS is implemented to detect network-based attacks on network devices. These attacks, often targeting protocol vulnerabilities, cause a significant threat. One example is Denial-of-Service (DoS) attacks, which aim to impact the availability of devices or networks [3, 37]. The huge volume of data can become a real concern, with recent research exploring dimensionality reduction and smart processing for efficient alert handling [18]. Additionally, trust-based schemes are being proposed to ensure data quality while reporting critical information [3, 6, 13, 37]. The emerging infrastructure, protocols, new attacks, and systems in computer networks demand careful consideration when designing an NIDS. Addressing complex data handling, huge volume of network traffic, misclassification issues and trust concerns is essential for effective intrusion detection in such networks. Researchers have explored and successfully implemented various ML and DL techniques to create anomaly-based NIDS. Many of these implementations still face the issue of misclassifying certain attack packets as legitimate ones, often resulting in critical consequences. This is particularly problematic in decision-making applications, where such errors can have severe implications on network security. To address this, we propose augmenting anomaly-based NIDS with a secondary layer of classification using ML techniques, incorporating zero-watermarking and data provenance.

3.2 Data Provenance

Data provenance is a concept that is applied in many research disciplines. Every application domain has a unique definition of provenance [25]. In network security, data provenance ensures the reliability and trustworthiness of data by tracking the ownership chain and actions performed on generated data from the source node to the final destination. Every data packet that is received from source nodes must have its provenance recorded, and forwarding nodes' involvement in the data transmission process must be tracked. However, implementing such a solution is a challenging task. The rapid growth of provenance data in computer networks during the transmission phase is one major problem. Furthermore, restrictions are imposed by the bandwidth, computational overhead and data storage capacities [16]. Data provenance ensures that the user trusts the data received at the final destination, confirming that the data is collected by the designated unique authorized device at the specified time and location [4]. In our model, we use provenance information, such as source IP address, destination IP address, packet sequence number, and timestamp, to generate zero-watermarks for data packets at each source device.

3.3 Zero-Watermarking

One of the most well-known developments in network security is digital watermarking. It can accurately identify whether data has been altered and successfully blocks data interception. Moreover, it can be applied to secure copyright data as well as the content integrity of digital multimedia works, including audio, video, and images [35]. Comparing digital watermarking to other security methods, there are several advantages, including the following:

1. Because watermarking requires few calculations, its three processes: generation, embedding, and extraction use less energy.
2. Watermark information is directly stored in carrier data without requiring additional network connection cost [7].
3. When compared to previous security solutions that need a high degree of complexity, digital watermarking significantly minimizes end-to-end delay because of the lightweight watermark generation procedure.

A relatively recent technique for digital watermarking is zero-watermarking. Watermark generation, embedding, and extraction processes vary depending on the type of watermarking technique. Examples include hash functions (cryptographic schemes), unique codes inserted in information-hiding schemes, and bit position modifications [10]. Zero-watermarking approaches involve the generation of watermarks by the source node by the extraction of significant features from the original data without modifying the data related to these features. Zero-watermarking allows for the application of various functions for the generation process. The generated watermarks in zero-watermarking are not embedded in the data payload, but it is invisibly added to the data packet and without any modification to the data payload. Although several zero-watermarking techniques exist in the literature, very few methods are proposed to ensure data integrity and secure provenance in network security. Furthermore, to the best of our knowledge, there are no proposed techniques that use zero-watermarking in an NIDS with data provenance. In Section 4, we present an in-depth description of our model, which augments ML with a zero-watermarking-based data provenance scheme to achieve high accuracy and minimize false alarm rate in attack classification.

4 PROPOSED MODEL

We propose a novel two-layer classification approach called ZW-IDS for intrusion detection to enhance the performance of NIDS. The approach is based on integrating a zero-watermarking-based scheme with an anomaly-based NIDS. The proposed approach is divided into two classification layers: (1) the first one is carried out by applying ML using SVM and feature engineering, and (2), in the second layer, the classification is performed on the classified data from the first layer using a zero-watermarking scheme with data provenance information. The workflow of the proposed model is given in Figure 1. The placement of the NIDS is at the gateway to capture the flow of data packets from source devices and incoming network traffic. The framework includes different components for processing data before applying the classification methodology. The proposed model is thoroughly discussed in this section.

4.1 Data Pre-processing Stage

This stage cleans and prepares data from the dataset for further analysis, training and testing. This includes handling missing values, outliers, formatting inconsistencies, data normalization and standardization. The pre-processing process in our work includes the following steps:

1. **Calculating ranges for numeric features:** This process calculates and returns the range, which is the difference between the maximum and minimum values, for each numeric feature within the dataset. Range calculation is very important to provide insight into the spread or variability of numerical data. By understanding the range of each numeric feature, we can assess the scale of the data and identify potential outliers or anomalies. It helps in determining the relative importance of features during analysis and ensures that the features are on a similar scale for the SVM model, thus preventing bias towards features with larger ranges. We notice that there are two features having infinite numbers in many data packets which will affect the model performance. Consequently, we drop these two features that are *Flow Bytes* and *Flow Packets*.
2. **Encoding categorical values and labels:** This process encodes categorical values and labels. For labels, we have two classes *attack* and *normal*. The dataset includes a label feature where normal traffic is labeled as *normal*, and any intrusion is labeled as *attack*. Encoding categorical variables converts non-numeric data into a numerical format understandable by SVM models. We drop some categorical values that are not useful in the classification procedure such as: *Flow ID*, *Source IP*, *Destination IP*, *Timestamp*, and *Label*. This results in a 78 numeric feature.
3. **Data standardization:** Also known as data scaling or normalization, is a pre-processing technique used to transform the numeric features of the dataset to have a mean of 0 and a standard deviation of 1. This process ensures that all features are on a similar scale, preventing features with larger magnitudes from dominating those with smaller magnitudes during model training. In our work, `StandardScaler()` from `scikit-learn` is used to standardize the data. This scaler calculates the mean and standard deviation for each feature and then scales each feature such that it has a mean of 0 and a standard deviation of 1. By standardizing the data before training our SVM model, we ensure that the decision boundary is not biased by features with larger scales. The standardization of these features is carried out using the following equation:

$$z_i = \frac{x_i - \mu_i}{\sigma_i},$$

where z_i is the standardized value of the i-th feature, x_i is an individual observation of the i-th feature, μ_i is the mean of

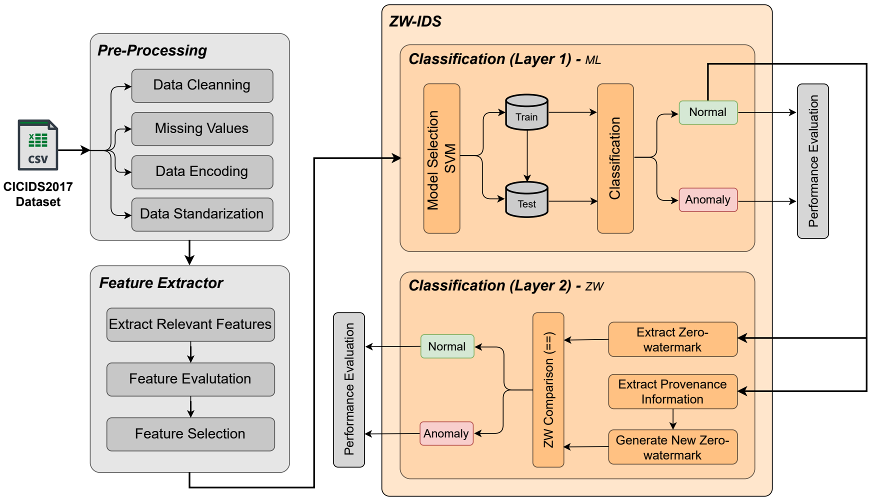

Figure 1: Proposed ZW-IDS workflow.

the i-th feature, and σ_i is the standard deviation of the i-th feature.

4.2 Zero-watermark Generation and Embedding

4.2.1 ***Provenance Information Extraction.*** Within the selected CICIDS2017 dataset, we encounter 85 data features, with 5 of them being non-numeric: source IP address, destination IP address, timestamp, flow ID, and label. These particular features are consistently removed by existing ML-based intrusion detection approaches during the data pre-processing phase, as they hold no relevance to the classification process. However, we recognize their significance as provenance information and choose to extract them for zero-watermark generation process. Initially, a sequence number, part of provenance information, is created for each packet, using extracted features as the source and destination IP addresses, timestamp, and flow ID. These accumulated provenance information, along with the generated sequence number, is used to generate the zero-watermark for each individual observation or data packet x.

4.2.2 ***Zero-watermark Generation and Embedding Procedure.*** Introducing a zero-watermarking approach to augment an ML-based IDS requires a watermark generation and embedding process at each legitimate source device. We propose a new zero-watermark generation and embedding algorithm to embed a watermark to the transmitted data packets using provenance information. Algorithm 1 describes the process of generating and embedding a watermark. It accepts data packets from the source device to generate a final zero-watermark. The algorithm extracts provenance information from the data features of each data point such as, *source IP address*, *destination IP address*, *timestamp* and combines it with a generated unique data packet sequence number (*seq*) to generate a sub-watermark $sw_{f_{n,k}}$ as shown in Algorithm 1. Then, the sub-watermark is encrypted using the Advanced Encryption Standard (AES). The input data is padded to ensure its length is a multiple of the AES block size. $sw_{f_{n,k}}$ is encrypted using the secret 128 bit key K_j to obtain a provenance record $p_{n,k} = E(sw_{f_{n,k}}, K_j)$. Another sub-watermark $sw_{h_{n,k}}$ is generated from the hash value of data payload using a one-way hash function, SHA-2. SHA-2 is preferred over other hash functions like MD5 due to its lightweight nature, consuming 65% less memory. MD5, while widely used in the past, has vulnerabilities that compromise its security [8]. Finally, these two generated sub-watermarks are concatenated to form a final zero-watermark $W_{F_{n,k}}$ using the following equation:

$$W_{F_{n,k}} = E(w_{ip} \,||\, w_t \,||\, w_{sq}, K_j) \,||\, H(x_{n,k}) = E(sw_{f_{n,k}}, K_j) \,||\, sw_{h_{n,k}}.$$

where $W_{F_{n,k}}$ $(1 \leq n \leq N)$, is the final watermark, N is the number of devices in the network, $||$ denotes the concatenation operator, H is a secure and lightweight one-way hash function, and n is the source device number. After the generation procedure, the final zero-watermark is embedded in the data packet x as shown in Equation 1 and, then, undergoes transmission.

$$x_{(n,k)W_{F_{n,k}}} = x_{n,k} \,||\, W_{F_{n,k}}. \tag{1}$$

4.3 Feature Extraction and Selection

Network connections can be described using a collection of data features, but these vary in their impact for understanding the connection's behavior. Some features provide little to no relevant information and are considered irrelevant. Others contain repetitive data and are redundant [29]. For this, we use a feature extraction and selection procedure to identify and extract relevant features from the pre-processed data that are informative for anomaly detection, and evaluate and select the most important features. We test the different SVM models on different feature selection procedures. The two main cases are: (1) using all dataset features, and (2) selecting the k-important features using the ensemble learning method `ExtraTreesClassifier`. The method is as follows:

- **`ExtraTreesClassifier():`** Extra Trees, which stands for Extremely Randomized Trees, is an ensemble learning method based on decision trees. It creates a forest of random decision trees and splits nodes using random thresholds. This randomness helps to reduce overfitting and variance in the

Algorithm 1 : Watermark Generation and Embedding

```
input: $x_{n,k}$
output: $W_{F_{n,k}}$
procedure Watermark Generation
    $w_{ip_s} \leftarrow$ network device $n$ IP Address
    $w_{ip_d} \leftarrow$ destination device IP Address
    $w_t \leftarrow$ extracted timestamp $(x_{n,k})$
    $w_{sq} \leftarrow$ packet sequence number $(seq(x_{n,k}))$
    $sw_{f_{n,k}} \leftarrow w_{ip_s} \mid\mid w_{ip_d} \mid\mid w_t \mid\mid w_{sq}$
    $p_{n,k} \leftarrow E(sw_{f_{n,k}}) \leftarrow \mathrm{ENC}(K_j, sw_{f_{n,k}})$
    $sw_{h_{n,k}} \leftarrow H(x_{n,k})$        ▷ select first 8 bytes of hash output
    $W_{F_{n,k}} \leftarrow E(sw_{f_{n,k}}) \mid\mid sw_{h_{n,k}}$
end procedure
procedure Watermark Embedding
    $x_{(n,k)W_{F_{n,k}}} \leftarrow x_{n,k} \mid\mid W_{F_{n,k}}$
    $Send\Big(x_{(n,k)W_{F_{n,k}}}\Big)$
end procedure
```

model. We train the model on the input features (78 features) of the CICIDS2017 dataset and the target variable which is the labels (normal, attack). In this step, the model's parameters is adjusted so that it can map the input data to the correct output labels.

- After training the model, the feature importance scores is calculated. Feature importance indicates the relative importance of each feature in predicting the target variable. The model returns an array containing the importance scores for each feature. We select the features that affect the decision of mapping each data point to a target label. Thus, we suppress the features with the least important scores (which are 30) resulting in 48 features.

4.4 Classification

This section describes the two-layered approach for intrusion detection. The first layer uses an SVM classifier to identify anomalies in the network traffic. The second layer leverages a zero-watermark approach, which uses provenance information to be embedded within the data packets. This layer adds an extra layer of security by extracting important information to verify data integrity and further enhance intrusion detection performance.

4.4.1 *Classification (Layer 1) using ML.*

An SVM is a supervised learning algorithm that excels at separating data packets into two categories, which is a generalization of maximal margin classifier. It sets a dividing line (hyperplane) in a multidimensional space. SVM aims to find the maximum margin (distance) between this hyperplane and the closest data points (support vectors) from each category.

In our network dataset, we can represent it as an $n \times p$ data matrix X, which includes n training data packets in a $p-$dimensional feature space $x_1 = \begin{pmatrix} x_{11} & \cdots & x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} & \cdots & x_{np} \end{pmatrix}$. These data packets belong to two main classes, $y_1, \ldots, y_n \in \{-1, 1\}$, where -1 represents attack class and 1 normal class. A new data packet is received, a $p-$vector with data features $x^* = (x_1^*, \ldots, x_p^*)^{\mathrm{T}}$. Our objective is to construct a classifier using our training dataset, enabling the classification of incoming data packets based on its set of features.

The support vector classifier is effective for linear classification in a two-class scenario, but real-world boundaries are often nonlinear. SVM extends this by enlarging the feature space using kernels. Solving the SVM problem relies on the inner products of of the data points. The inner product appears every time in the representation or the calculation of the solution for the support vector classifier, for that it is replaced with a generalization of the inner product of the following form: $K(x_i, x_{i'})$, where K is a function called *kernel.* There are several kernels that can be used for classification in SVM method such as *linear*, *polynomial*, *radial*, and *sigmoid.* In our approach, we train these four models on labeled data to learn a decision boundary between normal and anomaly packets based on the selected features in Section 4.3. Then, we apply the trained model to classify new, unlabeled data packets as normal or anomaly. After classification, the classified normal packets are placed in a CSV file to be used as an input to the next classification procedure. This file holds the predicted normal packets based on the specified decision boundary by the SVM model. In this stage, we apply all possible SVM models to test which one gives the best performance in terms of accuracy, precision, recall, F-score, computational performance and highest Area Under Curve (AUC) in the Receiver Operating Characteristic (ROC) curve. This is carried out by applying a *GridSearchCV* for testing the best SVM model using 5−fold cross-validation and get the best parameters C and γ. After obtaining the highest performance from parameter tuning, we also apply Principal Component Analysis (PCA) to reduce the dimentionality of the feature space and check the performance using two PCAs. After extensive experiments, we found that the best performance is obtained using an RBF kernel with $C = 100$ and $\gamma = 0.1$. The detailed results of these experiments are beyond the scope of this paper due to space limitations. The RBF uses the following function for attack classification:

$$K\left(x_i, x_{i'}\right) = \exp\left(-\gamma \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2\right).$$

where γ is a positive constant parameter that determines the influence of each training sample on the model. It defines the reach of the kernel function, controlling the flexibility of the decision boundary.

4.4.2 *Classification (Layer 2) using Zero-Watermarking and Data Provenance.*

In the second layer classification, we use the best model and parameters of SVM that we already tested over the network dataset which is the RBF kernel. The aim of this layer is to check the misclassified data packets which are actual attack packets but are classified as "normal" ones. This is the Type 1 error in the classification process which is misclassifying a sample that belongs to the attack class as belonging to the normal class. In other words, it is a false positive. In network security applications, the attack class represents any type of an intrusion detected within the devices or network, while the normal class represents regular or expected behavior. Reducing Type 1 errors is critical in such applications, because missing an actual attack can lead to significant consequences overwhelming targeted servers, devices or networks.

In our model, we generate a zero-watermark at the source device and embed it in the data packet before transmission using extracted provenance information, as shown in Algorithm 1.

After applying SVM classification in the first layer, the classified normal packets are used as an input to the zero-watermark algorithm to detect whether a packet is misclassified as normal or it is an actual normal packet. After receiving the initially flagged normal packet $x'_{(n,k)W_{F_{n,k}}}$, we extract provenance information (w_{ip_s}, w_{ip_d}, w_t, w_{sq}) from data features and re-generate a new zero-watermark $R(W'_{n,k})$ based on Algorithm 2. The sequence number that we generated is based on flow ID, source IP, destination IP and timestamp to uniquely distinguish data packets from source devices. Then, we extract the zero-watermark $W_{F_{n,k}}$ from the data packet $x'_{(n,k)W_{F_{n,k}}}$. If both zero-watermarks are equal then it remains flagged as a normal packet. Otherwise, it is re-flagged as an anomaly. In this scenario, there are two possibilities: either the data packet undergoes modification in its payload or zero-watermark, or the attacker generates zero-watermarks using their own secret key, which won't be authenticated by the IDS. This is a misclassification from the SVM layer 1 model. However, if the received packet lacks a zero-watermark, it falls into one of two categories. First, it may be a control packet, which follows a different procedure which is outside the scope of this paper. Alternatively, if it does not meet the criteria of a control packet, it is classified as an intrusion. After the second layer of attack detection, we evaluate the performance by applying the same metrics that are used in the SVM model evaluation, as shown in Figure 1.

Algorithm 2 : Zero-watermark Re-generation and Re-classification

```
input: $x'_{(n,k)W_{F_{n,k}}}$
output: normal/attack
procedure WATERMARK RE-CLASSIFICATION
    $Receive\left(x'_{(n,k)W_{F_{n,k}}}\right)$
    $R(W'_{n,k}) \leftarrow$ REDO Algorithm 1
    $W_{F_{n,k}} \leftarrow$ extract $W_{F_{n,k}}$ from $x'_{(n,k)W_{F_{n,k}}}$
    if $W_{F_{n,k}} \notin x'_{(n,k)}$ then
        flag $x'_{(n,k)}$ as 'attack'
    elif $R(W'_{n,k}) = W_{F_{n,k}}$ then
        flag $x'_{(n,k)}$ as 'normal'
    else
        flag $x'_{(n,k)}$ as 'attack'
    end if
end procedure
```

5 EXPERIMENTAL EVALUATION

The ML and zero-watermarking algorithms implementation were done using Scikit-learn[1] library in Python[2] as backend. The code was developed in the Visual Studio™ environment using 16 GB RAM and an Intel™ i7 2.59 GHz processor. The experiments were carried out using CICIDS2017 [31] dataset.

5.1 Dataset Description

CICIDS2017 is a network traffic dataset designed for evaluating IDS. It includes both normal traffic and diverse cyberattacks, offering a realistic testing ground for IDS. This makes it suitable for training and testing IDS models, particularly for network environments where traditional attack scenarios often apply. The dataset includes a number of attacks such as Brute Force FTP, Brute Force SSH, DoS, Web Attack, Botnet and Distributed Denial-of-Service (DDoS). The dataset includes 2.8 million samples of network traffic. To make evaluation of our model feasible, we use the DDoS network traffic CSV file and split the dataset into 70% training and 30% testing samples. The chosen dataset shows a good balance between normal and attack packets and includes 85 data features extracted from network traffic. Table 1 shows an overview of the classes within the dataset.

Table 1: Overview on CICIDS2017 dataset.

Dataset type	Number of data samples		
	Records	Normal	Attack
CICIDS2017 Train	158021	68311	89710
	%	43.22	56.77
CICIDS2017 Test	67724	29407	38317
	%	43.42	56.57

5.2 Evaluation Metrics

The performance metrics defined in Table 2 are used to evaluate the approach. These include accuracy, precision, recall, F-score, false negative rate (FNR), false positive rate (FPR) and computational time. These metrics are are defined from True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN) values. TP represents correctly identified positive cases, FN denotes negative cases incorrectly labeled as positive, FP indicates positive cases incorrectly labeled as negative, and TN signifies correctly identified negative cases.

5.3 Results and Discussion

5.3.1 ***Performance Evaluation on the CICIDS2017 Dataset***. We have conducted several experiments to explore various scenarios involving SVM configurations, feature selection methods, and hyper-parameter adjustments with our proposed zero-watermarking classification using provenance information.

In our initial scenario, we built an IDS using all features from the CICIDS2017 dataset, incorporating our novel zero-watermarking classification layer. We evaluated different combinations of our proposed ZW-IDS model with different SVM kernels, including linear, RBF, polynomial, and sigmoid. Using *GridSearchCV*, we optimized the hyper-parameters (C, γ, d) to achieve the best performance in terms of classification metrics and computational efficiency. Notably, we found that setting hyper-parameters to $C = 100$, $\gamma = 0.1$,

[1] https://scikit-learn.org/
[2] https://python.org/

Table 2: Performance metrics.

Metric	Explanation
$\text{Accuracy} = \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$	Overall ratio of correct predictions made by the model.
$\text{Precision} = \dfrac{\text{TP}}{\text{TP} + \text{FP}}$	Ratio of positive predictions that are actually correct.
$\text{Recall} = \dfrac{\text{TP}}{\text{TP} + \text{FN}}$	Ratio of actual positive cases that were identified correctly.
$\text{F-score} = \dfrac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	Harmonic mean of precision and recall.
$\text{FNR} = \dfrac{\text{FN}}{\text{FN} + \text{TP}}$	Ratio of actual positive cases that were incorrectly classified as negative.
$\text{FPR} = \dfrac{\text{FP}}{\text{FP} + \text{TN}}$	Ratio of actual negative cases that were incorrectly classified as positive.

Table 3: Performance evaluation with existing approaches.

Approach	Year	Classification Performance Metrics (%)						Computational Time (s)	
		Accuracy	Precision	Recall	F-score	FPR	FNR	Training Time	Testing Time
Tao et al. [33]	2018	98.01	98.39	98.16	98.27	1.89	2.06	49953.13	3.13
Jan et al. [11]	2019	98.03	98.43	97.99	98.20	1.91	2.01	208.90	2.28
Ravi et al. [27]	2022	98.77	97.84	98.74	98.21	1.18	1.34	–	–
Rao and Suresh Babu [26]	2023	98.97	99.06	98.17	99.73	3.93	5.02	–	–
Alghushairy et al. [1]	2024	88.74	–	98.82	–	12.19	1.17	–	**0.007**
Proposed ZW-IDS	2024	**99.98**	**100.0**	**99.96**	**99.97**	**0.0**	**0.034**	**8.1**	4.8

(–): *Performance metric is not reported in the approach.*

and $d = 3$ with the RBF kernel, combined with zero-watermarks generated using AES encryption with a 128-bit secret key, show the best results. This configuration achieved an accuracy of 98%, with training and testing times of 9.8 and 2.5 seconds, respectively. Additionally, we applied dimensionality reduction using PCA to reduce the feature space from 78 dimensions to 2 components. However, the integration of PCA led to a decline in classification performance, reducing accuracy to 96% and increasing training and testing times to 70.2 and 49.5 seconds, respectively. Thus, our findings suggest that in our context, dimensionality reduction does not effectively improve classification accuracy or reduce computational overhead.

In the second scenario, we employed feature selection using the ExtraTreesClassifier ensemble learning model to assess the importance scores of all features within the CICIDS2017 dataset. We identified and removed 30 unimportant features that did not significantly impact the classification process between normal and attack classes. The number of selected features not only enhanced processing time but also improved classification performance. Repeating experiments under the same conditions as the first scenario, with optimal hyper-parameter values of $C = 100$, $\gamma = 0.1$, and $d = 3$, along with the same zero-watermark generation and verification processes, shows better performance results. Applying feature selection with the zero-watermark approach using the RBF kernel resulted in a significant performance improvement, achieving a 99.98% accuracy and an very low false alarm rate of 0.034%. The classification results of data packets, categorized into two classes –'normal' and 'attack'– using our approach, are shown in the confusion matrix presented in Figure 2. Moreover, the proposed model achieve better computational efficiency, resulting in 8.1 seconds in training time and 4.8 seconds in testing time (including zero-watermark regeneration and verification time). Thus, integrating feature selection and hyper-parameter tuning with our proposed ZW-IDS effectively mitigates model biasing and overfitting issues, enhancing the effectiveness and speed of IDS in detecting attacks and minimizing misclassification of data packets.

5.3.2 *Comparison of Proposed ZW-IDS with Existing Approaches*.

To assess the effectiveness of our approach, we conducted a performance comparison with five state-of-the-art methods proposed by Tao et al. [33], Jan et al. [11], Ravi et al. [27], Rao

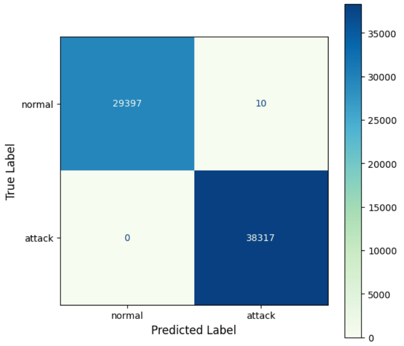

Figure 2: Confusion Matrix. A confusion matrix depicting our proposed ZW-IDS classification model's performance, where true positives (TP) are in the upper left corner, true negatives (TN) in the lower right, false negatives (FN) in the upper right, and false positives (FP) in the lower left.

and Suresh Babu [26], and Alghushairy et al. [1] in terms of classification performance and computational efficiency, as detailed in Table 3. Our approach demonstrates better performance, achieving the highest accuracy of 99.98%, precision of 100%, recall of 99.96%, and F-score of 99.97%. These results shows the effectiveness of introducing a second-layer attack detection based on data provenance (zero-watermark) to augment an ML-based IDS.

While other methods employ various feature selection techniques with SVM, such as genetic algorithms proposed by Tao et al. [33], focusing on specific attributes like packet arrival rate as suggested by Jan et al. [11], or combining SVM with GNB presented by [1], our approach consistently outperforms them. Even when compared to other ML algorithms like Naive Bayes, Logistic Regression, KNN, DT, Random Forest proposed by Ravi et al. [27], as well as DL models like RNN, LSTM, and GRU proposed by [26], our proposed ZW-IDS approach demonstrates better performance. Furthermore, ZW-IDS achieves significantly lower false error rates, with an FNR of 0.034% and no instances of Type 1 errors misclassifying attack data packets as normal packets as shown in the confusion matrix of Figure 2.

In terms of computational efficiency, while not all approaches consider this metric, our approach outperforms Tao et al. [33] and Jan et al. [11] in training time, offering significantly lower processing time with 8.1 seconds compared to both approaches with 49953.13 and 208.9 seconds, respectively. Although Alghushairy et al. [1] report a testing time of 0.007 seconds, their approach's classification performance falls short, with an accuracy of 88.74% and an FPR of 12.19%. In ZW-IDS, testing time includes the computational overhead of zero-watermark regeneration and verification procedure of layer 2 classification. This highlights the lightweight, effective, and efficient nature of our two-layer IDS for intrusion detection in the CICIDS2017 dataset. Moreover, we demonstrate that integrating zero-watermarking with data provenance in ML-based IDS enhances performance and facilitates effective intrusion detection while minimizing misclassification errors.

5.3.3 ***Effectiveness of Two-layered Approach.*** Combining zero-watermarking-based provenance technique with a ML-based IDS to form a two layer intrusion detection approach provide an effective solution to a number of issues that can not be achieved when applying only one layer of the model. These issues are as follows:

- The analysis of the dataset reveals a significant portion of attack packets as shown in Table 1, comprising nearly 57% of the total dataset. Applying SVM followed by the zero-watermarking model means only 43% of the data undergoes reclassification with zero-watermarking. In this case, if we want to only apply the second layer, it requires additional computational time due to the need for regeneration of zero-watermarks for each packet and subsequent comparison across the entire dataset. However, SVM classification layer mitigates this computational burden by classifying the majority of the dataset (57%).
- A critical concern arises if an attacker launches an attack from a compromised device and gains access to the AES encryption secret key used in the zero-watermark generation process. In such a scenario, the attacker can execute attacks with legitimately generated zero-watermarks, thereby bypassing detection at the IDS level. However, with the inclusion of the first layer, a significant number of attack packets can be identified prior to undergoing zero-watermark checks. Removing this initial layer and given knowledge of the secret key, the IDS would fail to detect any attacks.
- Another important consideration is that not all attacks can be effectively detected using zero-watermarking-based provenance data alone. The first layer of the IDS uses network traffic information to infer deviations from normal behavior in packet classification, a capability not inherently present in the zero-watermarking provenance solution and can not be achieved through relying only on the second layer.

The augmentation between both layers enhances the robustness and efficacy of the IDS, enabling effective detection of most attacks while minimizing computational overhead and improving classification performance.

6 CONCLUSION

This paper presents a novel approach to enhance the performance of anomaly-based NIDS by integrating zero-watermarking using data provenance information and an ML-based approach. The proposed approach addresses the limitations of traditional security measures by using ML techniques to differentiate between normal and malicious network activity. Through the implementation of SVM with feature selection in the first layer and data provenance-based zero-watermarking in the second layer, our method aims to reduce false alarms, improve computational efficiency, and enhance classification accuracy. Evaluation using the CICIDS2017 dataset shows the effectiveness of our approach in terms of classification performance and computational overhead. Additionally, a comparative analysis with existing multi-method ML and DL solutions highlights the improvement of our scheme in detecting and mitigating network intrusions. Overall, our proposed model contributes to advancing the field of network security by providing a practical and efficient solution for detecting and preventing cyber-attacks in information systems and computer networks.

Perspectives of future include continuous updating of the NIDS model, to effectively detect new and emerging forms of network attacks. The effectiveness of the proposed approach could also be evaluated in diverse network environments and under different attack scenarios. Additionally, considering the increasing adoption of the Internet of Things (IoT) devices and the increase of interconnected systems, future research can focus on extending the proposed approach to address security challenges in IoT networks. Developing lightweight and efficient anomaly detection techniques designed for IoT environments could help mitigate security risks associated with these emerging technologies.

ACKNOWLEDGMENTS

The authors acknowledge the funding obtained by the Detection of fake newS on SocIal MedIa pLAtfoRms (DISSIMILAR) project from the EIG CONCERT-Japan with grant PCI2020-120689-2 (Government of Spain), the PID2021-125962OB-C31 "SECURING" project funded by the Ministerio de Ciencia e Innovación, the Agencia Estatal de Investigación and the European Regional Development Fund (ERDF), as well as the ARTEMISA International Chair of Cybersecurity and the DANGER Strategic Project of Cybersecurity, both funded by the Spanish National Institute of Cybersecurity through the European Union – NextGenerationEU and the Recovery, Transformation and Resilience Plan. The authors acknowledge as well support from IMT's CyberCNI chair (Cybersecurity for Critical Networked Infrastructures, cf. https://cybercni.fr/).

REFERENCES

[1] Omar Alghushairy, Raed Alsini, Zakhriya Alhassan, Abdulrahman A. Alshdadi, Ameen Banjar, Ayman Yafoz, and Xiaogang Ma. 2024. An Efficient Support Vector Machine Algorithm Based Network Outlier Detection System. *IEEE Access* 12 (2024), 24428–24441. https://doi.org/10.1109/ACCESS.2024.3364400

[2] Juan Ignacio Iturbe Araya and Helena Rifà-Pous. 2023. Anomaly-based cyberattacks detection for smart homes: A systematic literature review. *Internet of Things* 22 (2023), 100792. https://doi.org/10.1016/j.iot.2023.100792

[3] Luca Arnaboldi and Charles Morisset. 2021. A Review of Intrusion Detection Systems and Their Evaluation in the IoT. https://doi.org/10.48550/arXiv.2105.08096 arXiv:2105.08096 [cs.CR]

[4] Elisa Bertino. 2018. *Security and Privacy in the IoT*. Springer, Cham, 3–10. https://doi.org/10.1007/978-3-319-75160-3_1

[5] Anna L. Buczak and Erhan Guven. 2016. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials* 18, 2 (2016), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[6] Christian Cervantes, Diego Poplade, Michele Nogueira, and Aldri Santos. 2015. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 606–611. https://doi.org/10.1109/INM.2015.7140344

[7] Chuhong Fei, Deepa Kundur, and Raymond H Kwong. 2006. Analysis and design of secure watermark-based authentication systems. *IEEE transactions on information forensics and security* 1, 1 (2006), 43–55. https://doi.org/10.1109/TIFS.2005.863505

[8] Prasanth Ganesan, Ramnath Venugopalan, Pushkin Peddabachagari, Alexander Dean, Frank Mueller, and Mihail Sichitiu. 2003. Analyzing and Modeling Encryption Overhead for Sensor Network Nodes. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications* (San Diego, CA, USA) *(WSNA '03)*. Association for Computing Machinery, New York, NY, USA, 151–159. https://doi.org/10.1145/941350.941372

[9] Muhammed Zekeriya Gündüz and Resul Das. 2019. Analysis of Cyber-Attacks in IoT-based Critical Infrastructures. *INTERNATIONAL JOURNAL OF INFORMATION SECURITY SCIENCE* 8, 4 (2019). http://search/yayin/detay/383257

[10] Khizar Hameed, Abid Khan, Mansoor Ahmed, Alavalapati Goutham Reddy, and M Mazhar Rathore. 2018. Towards a formally verified zero watermarking scheme for data integrity in the Internet of Things based-wireless sensor networks. *Future Generation Computer Systems* 82 (2018), 274–289. https://doi.org/10.1016/j.future.2017.12.009

[11] Sana Ullah Jan, Saeed Ahmed, Vladimir Shakhov, and Insoo Koo. 2019. Toward a Lightweight Intrusion Detection System for the Internet of Things. *IEEE Access* 7 (2019), 42450–42471. https://doi.org/10.1109/ACCESS.2019.2907965

[12] Seung-Ho Kang and Kuinam J. Kim. 2016. A feature selection approach to find optimal feature subsets for the network intrusion detection system. *Cluster Computing* 19, 1 (01 Mar 2016), 325–333. https://doi.org/10.1007/s10586-015-0527-8

[13] Zeeshan Ali Khan and Peter Herrmann. 2017. A Trust Based Distributed Intrusion Detection Mechanism for Internet of Things. In *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. 1169–1176. https://doi.org/10.1109/AINA.2017.161

[14] Chie-Hong Lee, Yann-Yean Su, Yu-Chun Lin, and Shie-Jue Lee. 2017. Machine learning based network intrusion detection. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*. 79–83. https://doi.org/10.1109/CIAPP.2017.8167184

[15] Hung-Jen Liao, Chun-Hung Richard Lin, Ying-Chih Lin, and Kuang-Yuan Tung. 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications* 36, 1 (2013), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

[16] Hyo-Sang Lim, Yang-Sae Moon, and Elisa Bertino. 2010. Provenance-Based Trustworthiness Assessment in Sensor Networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks* (Singapore) *(DMSN '10)*. Association for Computing Machinery, New York, NY, USA, 2–7. https://doi.org/10.1145/1858158.1858162

[17] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, and M.A. Zissman. 2000. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, Vol. 2. 12–26 vol.2. https://doi.org/10.1109/DISCEX.2000.821506

[18] Liqun Liu, Bing Xu, Xiaoping Zhang, and Xianjun Wu. 2018. An intrusion detection method for internet of things based on suppressed fuzzy clustering. *EURASIP Journal on Wireless Communications and Networking* 2018, 1 (09 May 2018), 113. https://doi.org/10.1186/s13638-018-1128-z

[19] Jun Luo, Senchun Chai, Baihai Zhang, Yuanqing Xia, Jianlei Gao, and Guoqiang Zeng. 2020. A novel intrusion detection method based on threshold modification using receiver operating characteristic curve. *Concurrency and Computation: Practice and Experience* 32, 14 (2020), e5690. https://doi.org/10.1002/cpe.5690

[20] Sergi Martinez-Bea, Sergio Castillo-Perez, and Joaquin Garcia-Alfaro. 2013. Real-time malicious fast-flux detection using DNS and bot related features. In *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*. IEEE, 369–372. https://doi.org/10.1109/PST.2013.6596093

[21] Inês Martins, João S. Resende, Patrícia R. Sousa, Simão Silva, Luís Antunes, and João Gama. 2022. Host-based IDS: A review and open issues of an anomaly detection system in IoT. *Future Generation Computer Systems* 133 (2022), 95–113. https://doi.org/10.1016/j.future.2022.03.001

[22] J. Mill and A. Inoue. 2004. Support vector classifiers and network intrusion detection. In *2004 IEEE International Conference on Fuzzy Systems (IEEE Cat. No.04CH37542)*, Vol. 1. 407–410 vol.1. https://doi.org/10.1109/FUZZY.2004.1375759

[23] Preeti Mishra, Vijay Varadharajan, Uday Tupakula, and Emmanuel S. Pilli. 2019. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Communications Surveys & Tutorials* 21, 1 (2019), 686–728. https://doi.org/10.1109/COMST.2018.2847722

[24] Usman Shuaibu Musa, Megha Chhabra, Aniso Ali, and Mandeep Kaur. 2020. Intrusion Detection System using Machine Learning Techniques: A Review. In *2020 International Conference on Smart Electronics and Communication (ICOSEC)*. 149–155. https://doi.org/10.1109/ICOSEC49089.2020.9215333

[25] Unkyu Park and John Heidemann. 2008. *Provenance in Sensornet Republishing*. Springer-Verlag, Berlin, Heidelberg, 280–292. https://doi.org/10.1007/978-3-540-89965-5_28

[26] Yamarthi Narasimha Rao and Kunda Suresh Babu. 2023. An Imbalanced Generative Adversarial Network-Based Approach for Network Intrusion Detection in an Imbalanced Dataset. *Sensors* 23, 1 (2023). https://doi.org/10.3390/s23010550

[27] Vinayakumar Ravi, Rajasekhar Chaganti, and Mamoun Alazab. 2022. Recurrent deep learning-based feature fusion ensemble meta-classifier approach for intelligent network intrusion detection system. *Computers and Electrical Engineering* 102 (2022), 108156. https://doi.org/10.1016/j.compeleceng.2022.108156

[28] Fadi Salo, Mohammadnoor Injadat, Ali Bou Nassif, Abdallah Shami, and Aleksander Essex. 2018. Data Mining Techniques in Intrusion Detection Systems: A Systematic Literature Review. *IEEE Access* 6 (2018), 56046–56058. https://doi.org/10.1109/ACCESS.2018.2872784

[29] Sobin Soniya Sathiyadhas and Maria Celestin Vigila Soosai Antony. 2022. A network intrusion detection system in cloud computing environment using dragonfly improved invasive weed optimization integrated Shepard convolutional neural network. *International Journal of Adaptive Control and Signal Processing* 36, 5 (2022), 1060–1076. https://doi.org/10.1002/acs.3386

[30] S. Seng, Joaquin Garcia-Alfaro, and Youssef Laarouchi. 2022. Why anomaly-based intrusion detection systems have not yet conquered the industrial market?. In

Foundations and Practice of Security - 14th International Symposium, FPS 2021, Paris, France, December 2021, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 13291). Springer Nature, 341–354. https://doi.org/10.1007/978-3-031-08147-7_23

[31] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,*. INSTICC, SciTePress, 108–116. https://doi.org/10.5220/0006639801080116

[32] Xu sheng Gan, Jing shun Duanmu, Jia fu Wang, and Wei Cong. 2013. Anomaly intrusion detection based on PLS feature extraction and core vector machine. *Knowledge-Based Systems* 40 (2013), 1–6. https://doi.org/10.1016/j.knosys.2012.09.004

[33] Peiying Tao, Zhe Sun, and Zhixin Sun. 2018. An Improved Intrusion Detection Algorithm Based on GA and SVM. *IEEE Access* 6 (2018), 13624–13631. https://doi.org/10.1109/ACCESS.2018.2810198

[34] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. 2009. Intrusion detection by machine learning: A review. *Expert Systems with Applications* 36, 10 (2009), 11994–12000. https://doi.org/10.1016/j.eswa.2009.05.029

[35] Ron G Van Schyndel, Andrew Z Tirkel, and Charles F Osborne. 1994. A digital watermark. In *Proceedings of 1st international conference on image processing*, Vol. 2. IEEE, 86–90. https://doi.org/10.1109/ICIP.1994.413536

[36] Emmanouil Vasilomanolakis, Shankar Karuppayah, Max Mühlhäuser, and Mathias Fischer. 2015. Taxonomy and Survey of Collaborative Intrusion Detection. *ACM Comput. Surv.* 47, 4, Article 55 (may 2015), 33 pages. https://doi.org/10.1145/2716260

[37] Kuldeep Yadav and Avinash Srinivasan. 2010. iTrust: an integrated trust framework for wireless sensor networks. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (Sierre, Switzerland) *(SAC '10)*. Association for Computing Machinery, New York, NY, USA, 1466–1471. https://doi.org/10.1145/1774088.1774402

[38] Yuan Zhang, Qinghai Yang, Sangarapillai Lambotharan, Konstantinos Kyriakopoulos, Ibrahim Ghafir, and Basil AsSadhan. 2019. Anomaly-Based Network Intrusion Detection Using SVM. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. 1–6. https://doi.org/10.1109/WCSP.2019.8927907

[39] Michael Zipperle, Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. 2022. Provenance-based Intrusion Detection Systems: A Survey. *ACM Comput. Surv.* 55, 7, Article 135 (dec 2022), 36 pages. https://doi.org/10.1145/3539605