information

MDPI

Article

Efficient Detection of XSS and DDoS Attacks with Bent Functions

Shahram Miri Kelaniki * and Nikos Komninos

School of Science and Technology, City St George's, University of London, London ECV 0HB, UK; nikos.komninos.1@citystgeorges.ac.uk
* Correspondence: shahram.miri-kelaniki@citystgeorges.ac.uk

Abstract

In this paper, we investigate the use of Bent functions, particularly the Maiorana–McFarland (M–M) construction, as a nonlinear preprocessing method to enhance machine learning-based detection systems for Distributed Denial of Service (DDoS) and Cross-Site Scripting (XSS) attacks. Experimental results demonstrated consistent improvements in classification performance following the M–M Bent transformation. In labeled DDoS data, classification performance was maintained at 100% accuracy, with improved Kappa statistics and lower misclassification rates. In labeled XSS data, classification accuracy was reduced from 100% to 87.19% to reduce overfitting. The transformed classifier also mitigated overfitting by increasing feature diversity. In DDoS and XSS unlabeled data, accuracy improved from 99.85% to 99.92% in unsupervised learning cases for DDoS, and accuracy improved from 98.94% to 100% in unsupervised learning cases for XSS, with improved cluster separation also being noted. In summary, the results suggest that Bent functions significantly improve DDoS and XSS detection by enhancing the separation of benign and malicious traffic. All of these aspects, along with increased dataset quality, increase our confidence in resilience detection in a cyber detection pipeline.

Keywords: Bent functions; machine learning; Maiorana–McFarland construction; DDoS; XSS

1. Introduction

Cyber-attack trends in 2025 show significant growth in combined Cross-Site Scripting (XSS) and Distributed Denial of Service (DDoS) threats targeting critical infrastructure and web applications. DDoS attacks have risen 39% year-over-year and surged by 54% in the second quarter of 2025. This rise in combined threats highlights the increased complexity and frequency of attacks against critical infrastructure agencies, government organizations, and private companies. These advanced adversaries use dynamic, polymorphic attacks that bypass traditional signature-based defenses and compromise data integrity, confidentiality, and availability. As these threats grow and become more complex, they need strong mathematical and cryptographic mechanisms capable of detecting subtle and abnormal changes in network behavior [1].

Although cross-site scripting (XSS) and DDoS attacks operate at different layers of the network, they share a common characteristic: the potential for evading detection through polymorphism and obfuscation. Conventional signature-based systems are not always able to address such dynamic threats, and conventional statistical characteristics may not be sufficient to distinguish between malicious and benign traffic. This study extends beyond

Academic Editors: Dharmaraj Veeramani and Jian Tang

Received: 4 December 2025
Revised: 23 December 2025
Accepted: 5 January 2026
Published:

standard feature engineering by leveraging the cryptographic properties of bent functions. In particular, we use the Maiorana–McFarland construction, which projects raw network properties into a very nonlinear space. We hypothesize that these functions are highly diffusive and nonlinear, thereby amplifying the small statistical aberrations in malicious traffic and improving machine learning models' classification accuracy.

Bent functions are a specific class of Boolean functions characterized by maximum nonlinearity and have demonstrated potential for enhancing cryptographic security and attack detection. Their nonlinear properties are well-suited for building hash functions, S-boxes, and learning-based intrusion detection systems, as they add unpredictability to the underlying mathematical structure. When bent function properties are integrated into machine learning environments, detecting cyber-attacks becomes more feasible, especially dynamic attacks like Distributed Denial of Service (DDoS) by recognizing variations in encrypted traffic [2].

Recent studies show that systems based on bent functions offer greater resilience against adaptive attackers and enhance the performance of algorithmically learned behaviors in anomaly detection models. By exploiting their sensitivity to initial conditions and their seemingly random behavior, bent functions can strengthen advanced detection and mitigation processes—all of which help develop next-generation cryptographic defenses and intelligent security management systems [2].

The importance of bent-function-based detection techniques is underscored by new encrypted and side-channel attack vectors; for example, harvesting radio-frequency (RF) energy from ambient Wi-Fi channels allows one to gain fine-grained inferences about mobile application activity via AppListener [3]. Such gains can be made without the need for packet capture or device compromise, instead relying on minute voltage fluctuations produced by application traffic behavior. This shows that even indirect means of obtaining behavioral information can expose app users; thus, there is a real demand for mathematically based nonlinear detection systems such as bent-function-based detection.

Application fingerprinting also exposes vulnerable user behavior in encrypted wireless traffic. The PACKETPRINT framework provides evidence that user behavior and application use can be inferred from packet characteristics (including packet size and directionality) earlier than previously realized, even in an unconstrained (open) environment with no visible boundaries or when multiplexed applications are used [4]. Sequential learning, through S-XGBoost, and hierarchical feature extraction, through H-BoW, enable PACKETPRINT to detect more sophisticated structures in packet captures that traditional methods cannot reliably identify. The results of this work provide practical support for developing models using high-entropy and nonlinear signal detection and illustrate the utility of bent functions for advanced anomaly detection.

Fine-Grained Open-World Android App Fingerprinting (FOAP) employs this technique to perform method-level inference for each user action on a device running an open-world Android app. The FOAP system uses structural similarity metrics to correlate network flows with Entry Point methods on the device, thereby excluding irrelevant network traffic and reducing false positives while maintaining behavioral resolution [5].

The combination of AppListener, PACKETPRINT, and FOAP illustrates the capability of modern cyber-attacks to exploit encrypted, side-channel, and polymorphic behaviors, which is a compelling reason to adopt a bent-function detection model as an architecturally correct foundation for robust, adaptive, and future-ready security systems.

A. Main Challenges and Motivation:

Traditional signature-based and conventional machine learning techniques often fall short in detecting zero-day and polymorphic attacks, as they are computationally intensive and lack adaptability while maintaining accuracy.

Given these challenges, this study will explore the applicability of bent functions, mathematical structures with maximal nonlinearity and balanced mappings. The properties mentioned above promote feature separation, protect valuable information, and provide discriminative power between benign and malicious network traffic. Citing the success of cryptography applications, bent functions will be utilized as innovative learning mechanisms to enhance detection accuracy and robustness. This approach aims to enable real-time operational effectiveness.

B. Contribution:

Our key contributions are as follows:

- Preprocessing DDoS and XSS data with the Maiorana–McFarland Bent function to amplify the non-linearity of features and representations and complement cyber-attack detection.
- Building a repeatable methodological scheme for applying Bent function transformations to network traffic data, enhancing feature discrimination and model interpretability.
- Decreasing overfitting based on the Bent function transformation, enhancing learning precision and the quality of clustering data, and providing data with an efficient mapping to a higher-dimensional space.
- Increasing the realization of substantial performance improvements with both supervised and unsupervised learning models with higher cluster separability and lower computational costs.
- Creating a method that demonstrates how Bent function transformations improve important network features for intrusion detection and can serve as a foundation for future systems.

C. Paper structure:

The remainder of this paper is organized as follows: In Section 2, we provide an overview of relevant related work on XSS and DDoS attacks. Section 3 introduces bent-function-based XSS and DDoS detection. Section 4 presents the experimental results and a discussion of them. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Related Work on XSS and DDoS Attacks

Cybersecurity research has increasingly concentrated on detecting and mitigating XSS and DDoS attacks through innovative, data-driven methods. These two types of attacks differ in nature (client-side code injection versus resource exhaustion), but they share common detection hurdles, such as obfuscation, dynamic payload creation, and scalability issues. Most current research utilizes machine learning and deep learning models to achieve higher accuracy and greater adaptivity than traditional rule-based or signature-based systems.

2.1. Cross-Site Scripting (XSS) Attack Detection

Cross-Site Scripting (XSS) is one of the most common client-side web vulnerabilities, enabling attackers to inject malicious scripts into trusted web environments. Traditional rule-based detection methods have proven ineffective against sophisticated obfuscation and polymorphic payloads. As a result, data-driven, machine-learning–based techniques for detecting XSS have gained increasing attention in recent research.

Initial methods for detecting Cross-Site Scripting (XSS) typically relied on rule-based, static analysis, which could easily miss payloads that were encoded or obfuscated in newer attacks. To overcome these limitations, authors in [6] published a study on identifying XSS vulnerabilities using an integrated statistical analysis with machine learning, proposing a hybrid framework to detect XSS payloads. The study employed Random Forest (RF) and

Support Vector Machine (SVM) classifiers—both with linear and polynomial kernels—to analyze encoded payloads represented as a Bag-of-Words (BoW). Results showed that data-driven models outperformed manual heuristics, especially in detecting XSS vulnerabilities, with RF achieving an F1 score of 99.69% and linear SVM reaching 99.66%. These findings demonstrated the effectiveness, scalability, and robustness of a machine learning–based approach for tackling XSS obfuscation, thereby enhancing web vulnerability detection and overall web security.

To address this, the authors in [7] proposed XSS Attack Detection Using Convolutional Neural Networks (CNN), which introduced a convolutional framework that facilitates learning of spatial and sequential patterns inherent in HTML payloads. By employing multi-level tokenization, n-gram segmentation, and convolutional feature extraction, the model presented in this research achieved a detection accuracy of 99.4% with a lower false-positive rate than RNN-based techniques.

To tackle the issue of limited labeled data for XSS detection, the authors in [8] introduced a few-shot learning framework titled "Research on Cross-Site Scripting Attack Detection Technology Based on Few-Shot Learning." The authors assembled a large dataset with over 210,000 samples. They tested architectures like CNNs, SVMs, and LSTMs to evaluate the performance of a few-shot learning setup. The authors also proposed a modified Virtual Sample Generation (VSG) algorithm to synthetically expand the training set, enhancing model generalization and robustness to unseen samples. Their CNN-based model achieved 0.99866 precision, 0.97593 recall, and 0.98716 F1-score; they then used the VSG-augmented model to reach an average accuracy of 78.6% on unseen CVE samples, compared to traditional bootstrap and perturbation methods. Although this approach was highly effective, the authors noted it could lead to significant computational overhead and be sensitive to drift over time.

In the same context, the authors in [9] introduced a Web Server Security Solution for the Real-time Detection of XSS Attacks using Deep Learning, implemented as a server-side detection system based on Multilayer Perceptron (MLP) architectures. The MLP model was trained on word embeddings with a dataset of 10,600 malicious payloads and 5000 benign samples, achieving 99.47% detection accuracy while defending against obfuscation methods based on HTML and JavaScript encoding.

In [10], a comparative analysis of Machine Learning Algorithms for XSS Detection evaluated four models (XGBoost, RF, KNN, and SVM). The RF model achieved the highest detection accuracy of 99.93% across various XSS payloads, demonstrating its effectiveness against a broad spectrum. An extension of this work is XSS-Net [11], a detection model employing TF-IDF and n-gram representation that incorporated a Logistic Regression (LR) with Principal Component Analysis (PCA) for dimensionality reduction, achieving 99.70% accuracy and 100% recall, outperforming the Random Forest model on payloads described as highly obfuscated.

The authors in [12] offer a broader perspective. They suggested that multiple Machine Learning Models should be used to detect XSS. This approach can reduce false positives and false negatives associated with existing rule-based defenses. The authors experimented with 138,569 real samples from trusted websites and public XSS repositories. They used these samples to train their RF, LR, SVM, CNN, ANN, and ensemble classifiers. Additionally, the researchers employed Feature Selection techniques based on Information Gain and ANOVA to reduce the number of features in the original dataset by 25%. The final results showed that Random Forest achieved a classification accuracy of 99.78%, and the ensemble models achieved an average accuracy of 99.76%. The study results demonstrate that the models accurately classified the majority of known attacks using the available data.

However, the study identified ongoing issues related to the availability of labeled data, feature evolution, overfitting control, and scalability for real-time processing.

The work was extended by [13], who emphasized speed and semantic understanding. Their approach used both Universal Sentence Encoder and Word2Vec embeddings to evaluate the contextual meaning of sentences and to generate a 612-dimensional feature space. The model was trained on 13,685 samples from PortSwinger and OWASP. The Random Forest classifier achieved 99.45% accuracy and identified each sample in under 1 ms. Although this design can be implemented in real time with minimal computational cost, it may not perform well when applied to unfamiliar attack styles or domains.

While such advancements are notable, it is clear that current systems remain vulnerable to evolving evasion techniques and mainly depend on large, labeled datasets. Given this reality, there is a strong need for lightweight, mathematically fair detection frameworks, such as bent-function–based nonlinear systems, that offer both strong detection resilience and computational efficiency. Overall, these studies mark a clear shift toward data-driven and hybrid learning approaches, emphasizing feature engineering, representation learning, and deep architectures as key drivers of improved accuracy, flexibility, and robustness against emerging internet-based threats.

2.2. Distributed Denial of Service (DDoS) Attack Detection

DDoS attacks continue to pose a serious risk to network infrastructure, especially in IoT and SDN networks. In DDoS detection, many researchers face the challenge of the increasing scale and complexity of network attacks. To identify DDoS patterns in IoT environments, the study in [14] introduced a hybrid deep learning model that combines CNN, LSTM, Autoencoder, and DNN architectures. Using the CIC-DDoS2019 dataset, their model achieved an overall accuracy of 80.75% and a processing time of 3.86 ms per instance, making it suitable for real-time intrusion detection.

To improve detection in Software-Defined Networks (SDN), Ref. [15] proposed an RNN–BiLSTM–based framework. The model was trained on both original datasets and adversarially generated datasets (using CTGAN), achieving a 98.08% F1-score on normal traffic and a 94.17% F1-score on adversarial traffic, demonstrating that deep models still face challenges from adversarial data.

A related work by the authors in [16], titled 'Feature Engineering and Machine Learning Framework for DDoS Detection in IoT,' enhanced the preprocessing pipeline using dimensionality reduction and Pearson correlation analysis. Focusing on the 6LoWPAN stack, the Random Forest classifier achieved 100% detection accuracy with zero false positives across all attacks launched via UDP flooding, demonstrating a lightweight, accurate solution suitable for low-resource IoT systems. To broaden this perspective, Ref. [17] examined Feature Identification Across IoT Layers for Dynamic Attack Detection. By establishing mappings between features at the perception, network, and application levels, the authors identified 20 important features using GainRatioAttributeEval. The J48 classifier attained 94.73% accuracy on these features, showing that it is feasible to extract layer-specific features for DDoS attacks on lower-cost, resource-constrained IoT devices.

The authors of [18] examined hardware-based detection methods that utilized Hardware Performance Counters (HPCs) to monitor low-level system events, such as instruction loads and cache references. Their kernel-level model achieved over 98% accuracy while decreasing feature dimensionality by 87.5%. The study shows that hardware metrics can be used for lightweight anomaly detection.

Lightweight ML methods have also been introduced in smart IoT environments. In [19], authors evaluated One-Class SVM (OCSVM) and Isolation Forest (IF) classifiers for anti-anomaly detection in simulated smart home networks. Their results show 96–

99% accuracy and low memory usage of around 15 MB, demonstrating the feasibility of the approach for real-time detection in resource-limited settings. Similarly, in [20], a multi-feature ensemble learning framework combining ANOVA F-score and Random Forest feature selection with a VotingClassifier of XGBoost and HistGradientBoosting for SDN-based DDoS detection was proposed, achieving 99.9% accuracy with virtually no false negatives. This work shows that ensemble models can offer both robustness and interpretability.

In [21], the researchers examined DDoS detection and mitigation for software-defined networks with centralized control planes, which make such networks more susceptible to distributed denial-of-service attacks, particularly in IoT and enterprise systems. A threshold-based framework was developed that integrates Flow for network traffic monitoring and OpenFlow for DDoS attack mitigation. The framework distinguishes between legitimate traffic and DDoS attack traffic by comparing flow rates against known bandwidth thresholds. In experiments using Hping3 to generate attack traffic against a 14-node topology, the threshold-based framework detected DDoS attacks in less than 3 s and kept attacking traffic below 1.25 Mbps. The framework offers low monitoring overhead and scalability and effectively mitigates low-rate SYN attacks. However, optimizing sampling rates and explicitly measuring false positives will require further investigation.

To implement a data-driven detection strategy, the researchers in [22] developed a hybrid CNN-LSTM framework to monitor both spatial and temporal trends in network traffic. This model, which underwent comprehensive preprocessing and feature selection on the CICDDoS2019 dataset, achieved 98.70% accuracy in distinguishing normal traffic from typical attacks, such as UDP and NTP amplification. Compared with the rule-based and threshold-based methods described in [21], the deep learning approach offers superior generalization and can model complex traffic patterns. However, it encountered challenges in accurately classifying DNS amplification attacks, achieving a productivity score of 0.76. These findings underscore the challenges posed by certain attack types and emphasize the need for further model customization and data diversification.

Despite these achievements, today's DDoS detection systems still face challenges in handling nonlinearity, complex feature interactions, and adaptive adversarial tactics. This prompts the exploration of function–based nonlinear frameworks that exhibit strong cryptographic properties and produce uniform output distributions, which can help improve stability and resilience in analyzing network traffic and detecting adversarial behaviors. Overall, advancements in XSS and DDoS detection demonstrate a trend toward integrating data-driven intelligence, such as machine learning and deep learning, within a more mathematically founded framework. In contrast, considerations of Bent functions, valued for their strong cryptographic properties and balance, provide principled support for enhancing interpretability, efficiency, and robust detection methods.

The distinguishing feature of the bent function, particularly in the Maiorana–McFarland type, is its ability to achieve greater nonlinearity and optimal balance, making it extremely difficult to attack or replicate. Robust cryptography can significantly enhance security against modifications such as polymorphic attacks, surpassing the capabilities of standard nonlinear methods.

Bent mapping not only modifies features. It also introduces a step of controlled cryptographic unpredictability, requiring the model to learn more than the underlying pattern. It adds another layer to the model's learning, where the model must understand how to work with the system's transformed data structure rather than relying solely on pattern recognition. As a result, bent mapping produces a more balanced distribution of training samples, reducing the potential for overfitting. With less overfitting, there is less similarity in how different models classify normal versus attack traffic. Finally, as clusters

are refined and made more specific through the bent mapping transformation, the overall accuracy of model predictions will improve compared to models trained solely on raw data.

3. XSS and DDoS Detection with Bent Functions

Research on bent functions (BFs) and generalized plateaued functions is well established in cryptography. BFs are Boolean functions with an even number of variables and constitute a proven subclass of Boolean functions with cryptographic importance. They possess the maximum possible degree of nonlinearity. Additionally, they exhibit the greatest Hamming distance from all affine functions, providing the highest resistance to linear approximations. The analysis of BFs is primarily spectral, and the Walsh–Hadamard transform is employed to evaluate spectral flatness in this context. The fundamental principle of optimized nonlinearity is essential in symmetric cryptography and coding theory to ensure resistance against structural and linear attacks. Current research concentrates on constructing BFs with the highest attainable algebraic degree. It also assesses properties such as algebraic immunity, which is another critical parameter in the design of cryptographic primitives [23].

Bent functions demonstrate both maximum nonlinearity and a balanced Walsh spectrum, offering strong protection against linear and differential cryptanalysis. Consequently, they help improve the detection of cross-site scripting (XSS) and distributed denial-of-service (DDoS) attacks in the Internet of Things (IoT) [24,25]. Bent functions are used to map network traffic and input data into highly nonlinear spaces within learning-based or cryptographic systems, making it challenging for adversaries to predict or bypass detection thresholds capable of obfuscation.

A balanced distribution of outcomes across all possible values enhances the performance of anomaly detection classifiers, especially when detecting obfuscated XSS payloads or encoded injections. This fair distribution ensures an even spread in feature space and reduces bias, thereby improving detection effectiveness [25]. The authors in [24] employ partition-based affine subspaces and high nonlinearity to generate large families of bent functions (over 2^{78} different 8-variable instances), greatly exceeding the scope of the completed Maiorana–McFarland class. Such interactions are particularly suitable for secure data transformations and privacy protection, especially in resource-limited systems.

In the field of DDoS mitigation, bent functions improve deep learning and federated anomaly detection models by introducing algebraic unpredictability during feature extraction. This trait helps better distinguish between legitimate and malicious traffic, while also reducing the influence of adversarial perturbations and correlations in clustering and classification [25,26]. Encoding with a bent function is more than a simple transformation; its nonlinearity exploits the algebraic structure to create non-linearly separable data patterns, aiding in the detection of stealthy, low-volume attacks across distributed networks [24,27]. Moreover, generalizations to cyclic groups (such as $\mathbb{Z}_2^k$-bent functions) enable efficient construction of nonlinear encodings from existing bent primitives and increase efficiency, allowing for real-time adaptive detection in IoT and edge computing environments [25].

Finally, bent functions offer a framework for creating secure, data-driven detection models that identify adaptive attacks like DDoS and XSS in highly resource-limited IoT environments.

3.1. Preliminaries and Background of Bent Functions

Bent functions were first introduced by Oscar Rothaus in the 1960s and officially published in 1976 to describe a specific class of Boolean functions that achieve maximum possible nonlinearity. A bent function reaches the highest possible Hamming distance from

any affine function, which means that bent functions are inherently resistant to various forms of cryptanalysis, especially linear cryptanalysis [28].

A Boolean function (BF) evaluates Boolean expressions by calculating contributions for n variables and is described as $F_n^2 \to F_2$. A truth table displays all possible outputs for each input combination, representing the BF:

$$f(x_1, x_2, \ldots, x_n) \to f(0,0,\ldots,0), f(0,0,\ldots,1), \ldots, f(1,1,\ldots,1) \quad (1)$$

This representation is the basis for the cryptanalysis and synthesis of Boolean functions [29].

In symmetric cryptography, Boolean functions are essential nonlinear elements used in designing pseudorandom generators, substitution boxes (S-boxes), and other critical parts of stream and block ciphers. Their algebraic and spectral properties provide confusion and diffusion, which are vital for protecting against attacks in linear and differential cryptanalysis [30].

3.1.1. Theoretical Foundations of Bent Functions

A fundamental property of bent functions in mathematics is that their Walsh–Hadamard transform has a constant magnitude of $\pm 2^{n/2}$. This shows that bent functions can only be defined for an even number of variables. The nonlinearity reaches its maximum level, making the secure design resistant to linear and differential cryptanalysis. This makes it a valuable primitive for constructing secure cryptographic components or for enhancing resilience in certain Boolean functions [28].

3.1.2. Construction of Bent Functions

A key focus in research on bent functions is their explicit construction. A common method involves combining one with a permutation to produce a new bent function. These new functions are often compared to existing families, such as the Maiorana–McFarland class, Dillon's Partial Spread class, and the Iterative Construction class, to demonstrate their novelty and independence [28].

A. Maiorana- McFarland (M-M):

The Maiorana–McFarland class is essential for construction. For a binary mapping of $f : \mathbb{F}_2^m \times \mathbb{F}_2^m \to \mathbb{F}_2$, with $n = 2m$ variables function is required. The M-M Bent bivariate function is one that can be described as:

$$f(x, y) = x \cdot \pi(y) + h(y) \quad (2)$$

where π represents a permutation and $h : \mathbb{F}_2^m \to \mathbb{F}_2$ is an arbitrary mapping, respectively.

The M-M construction is a traditional method for creating bent functions, but newer approaches, such as M-subspace uniqueness and bent 4-concatenations, also extend beyond it. Substitution boxes (S-boxes) based on the M-M construction can be reduced in size by up to 30% and are roughly 2.1 times faster than AES, thereby enhancing S-box efficiency for IoT [31].

B. Dillon's Partial Spread–Class:

The disintegration of this class support structure caused it to bend. The bending resulted from the collapse of his supporting framework. The bent function is:

$$F(x, y) = P\left(\frac{x}{y}, \frac{w}{z}\right) \quad (3)$$

where $P(x, y)$ is a balance function, and

$$(x, y) \in n \times n, \quad (w, z) \in m \times m \text{ mapping.}$$

The Partial Spread construction employs non-normal bent-degree-four functions and generates infinite families, providing designers with considerable flexibility for cryptographic designs. Research has shown an algebraic attack resistance of 99.8%, along with nearly a 4× reduction in timing side-channel leakage, making the construction appealing for secure embedded systems operating in adversarial environments [31,32].

C. Iterative construction:

The iterative construction for a two-variable Boolean function is given by:

$$f(x', x'') = p(x') + q(x'') \tag{4}$$

where both p and q are Bent functions.

Iterative methods work by gradually building complex bent functions through an additive process. They combine smaller parts of bent functions to create larger functions with controlled properties. This ability to control properties makes iterative methods especially useful in key scheduling and generating random outputs quickly. Evidence shows that libraries using these methods can achieve key generation and randomness up to 60% faster than SHA-3 with similar parameters, while still maintaining good information distribution and meeting NIST randomness standards [31,33].

4. Experimental Results and Discussion

4.1. Research Approach

We used the Cross-Site Scripting (XSS) dataset for deep learning, along with the CIC-DDoS2019 dataset available on Kaggle. It includes both labeled and unlabeled instances of DDoS and XSS attacks, serving as the empirical foundation for this study. Each dataset has 21 attributes and is processed using a specific construction of bent functions to meet the study's goals: (i) to evaluate the effectiveness of bent functions as a learning mechanism and (ii) to improve detection accuracy based on their mathematical features and properties.

Boolean functions are then analyzed to understand their properties that can be implemented in a learning environment, with the ultimate goal of protecting critical infrastructure and IoT devices. The evaluation of the detection mechanisms included the use of supervised and unsupervised algorithms.

The flowchart in Figure 1 illustrates how to select, consolidate, and learn attributes, which are essential for a methodological overview.

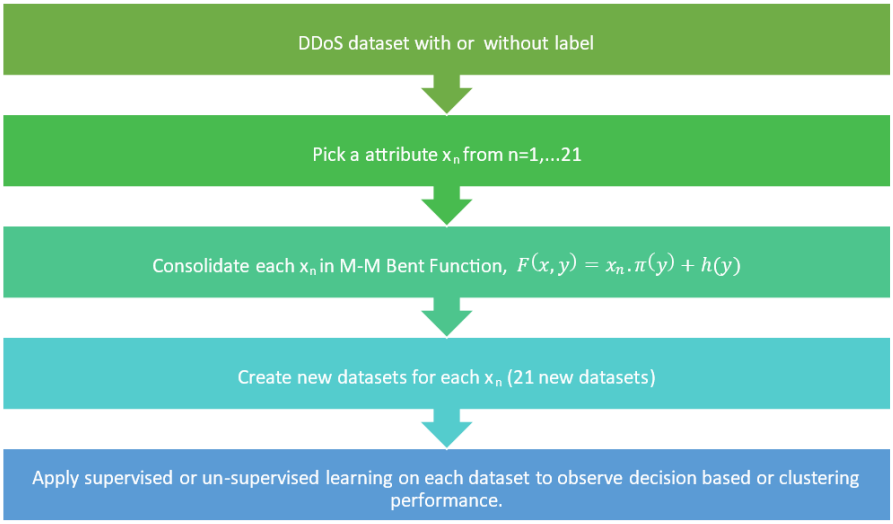

Figure 1. Process map to create a new dataset.

This research begins with a dataset obtained from a DDoS attack, which can be either labeled or unlabeled, depending on the analysis type. An individual attribute x_n, where $n = 1, \ldots, 21$, is selected from the dataset and applied to the M–M Bent Function

$$F(x, y) = x_n \cdot \pi(y) + h(y) \tag{5}$$

to transform that attribute. This process is repeated for all 21 attributes, resulting in 21 newly transformed datasets. Both supervised and unsupervised learning techniques are employed on each of the 21 transformed datasets to assess decision-making or clustering performance. This systematic approach enables researchers to investigate the impact of the M–M Bent transformation on dataset analysis used for detecting DDoS traffic.

4.2. Tools and Datasets

A mix of software tools and existing datasets is used in the study:

4.2.1. Tools

- Jupyter Notebook 6.5.7 (Python) was used for both data manipulation and implementing the Bent function, leveraging many libraries Python 3.13.3 provides for data processing and machine learning. It exports to either CSV or ARFF format, compatible with WEKA.
- WEKA GUI 3.8.9 was used to implement machine learning algorithms, visualize clustering patterns, and train and test data with those algorithms. The training-testing ratio can be adjusted from the default of 66–34%. Its interactive interface helps tune parameters and visualize data. The filter slider in the Weka Explorer interface is indicated by the green circle in Figure 2. Users can apply an unsupervised instance resampling technique to visually displayed DDoS test data using the filter slider.

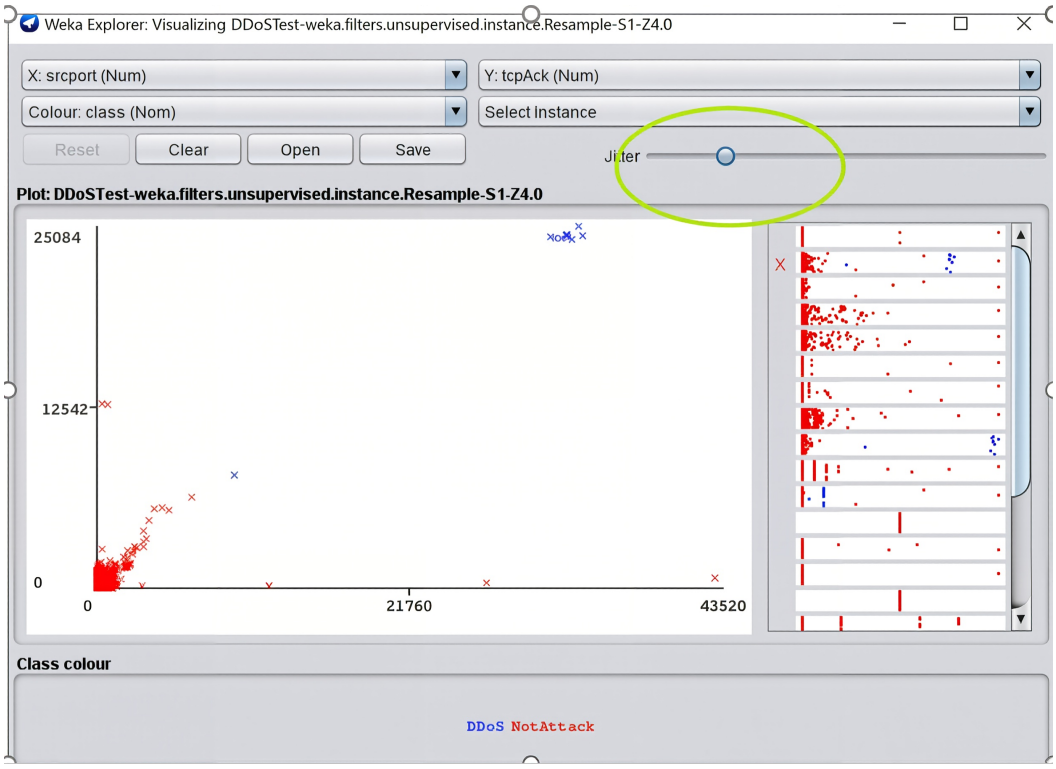

Figure 2. Scaling up visualization in WEKA GUI enabled better visualization.

4.2.2. Datasets Preparation

Using the labeled and unlabeled from Kaggle, which includes records of DDoS and XSS attacks (Table 1). As shown in Table 2, each dataset has 21 numerical attributes derived from network logs, and a 22nd attribute serves as the class label, indicating either Attack or NotAttack.

Table 1. Overview of the dataset.

Dataset	Labels	Size (Rows × Columns)	Primary Use
D1-DDoS, labelled	Yes	2059 × 22	Supervised Decision Tree
D2-DDoS, unlabelled	No	2059 × 21	K-Means clustering
D3-XSS, labelled	Yes	1800 × 22	Supervised Random Forest
D4-XSS, unlabelled	No	1800 × 21	Unsupervised clustering

Table 2. The Attributes of the Dataset.

Attribute	Variable	Description
download	X1	Related to the indicator for data downloads.
srcport	X2	The source port is a well-known port used for identifying and verifying port availability.
dstport	X3	This is the information about the destination port.
dnsEnquiries	X4	Request data from the DNS server.
dnsAnswers	X5	This is how a DNS server provides a response from its database.
ipAddressConnectTCP	X6	Connected to the TCP/IP communication protocol.
ipAddressConnectICMP	X7	The transmission of operational information to an IP address is associated with the Internet Control Message protocol.
pshAck	X8	It is required that the pushed buffered data is acknowledged.
noGet	X9	Access the GET request to filter web traffic.
noPost	X10	A part of the POST request.
noSYN	X11	Related to the byte sequence number of the of transmitted data.
requestContent	X12	An API that facilitates object creation.
postSize	X13	Concerned with the size of the POST method request.
ipBlackList	X14	Blacklisted IP address.
maliciousDNSIPConnect	X15	The malicious interference of a user's web browsing capability is responsible for this.
protocolTypeMal	X16	A protocol for detecting malicious hosts.
tcpFin	X17	The protocol indicated that the connection had ended.
packetLength	X18	This is the number of bytes that divide the current and next packets.
tcpAck	X19	The TCP protocol involves recording the last seen, sequence number of packets, expected sequence number, and Acknowledgement number.
analyseXSS	X20	Investigate the indicators of XSS.
analyseSQL	X21	Investigate the indicators in SQL.

4.2.3. Feature Extraction and Selection

First, a univariate statistical analysis using the Chi-Squared test was conducted to identify the most important features for distinguishing between attack and non-attack classes. In the DDoS dataset, the attributes x2 (srcport), x9 (noGet), x11 (noSYN), and x19 (tcpAck) showed the highest significance. Similarly, for the XSS dataset, the most relevant attributes were x2 (srcport), x9 (noGet), x11 (noSYN), x19 (tcpAck), and x20 (analyseXSS). Scatter plots of these attributes visually demonstrated their effectiveness in enhancing class separation and clearly defining the boundaries between attack and benign classes (see Figures 3–5).

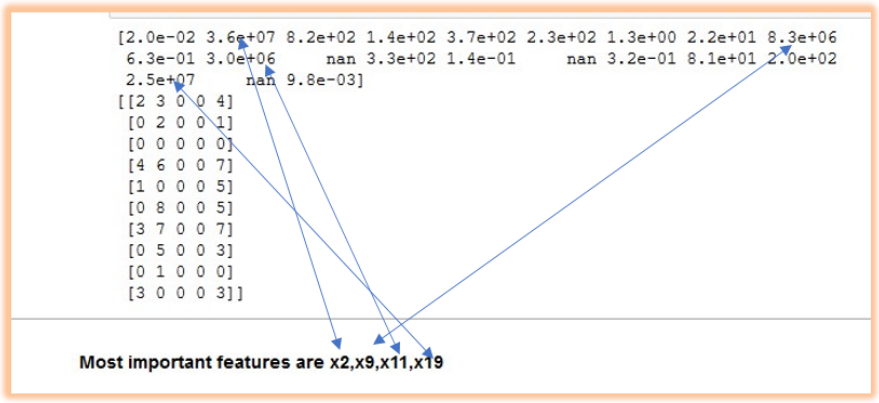

Figure 3. Feature selection on the DDoS labeled data.

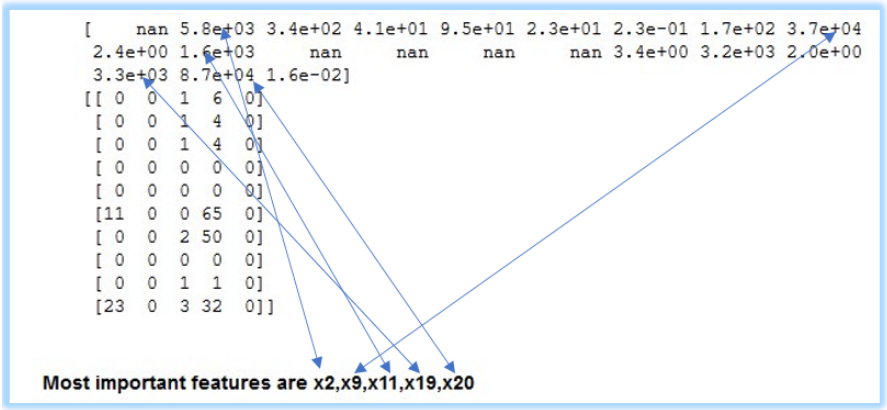

Figure 4. Feature selection applied to XSS-labeled data.

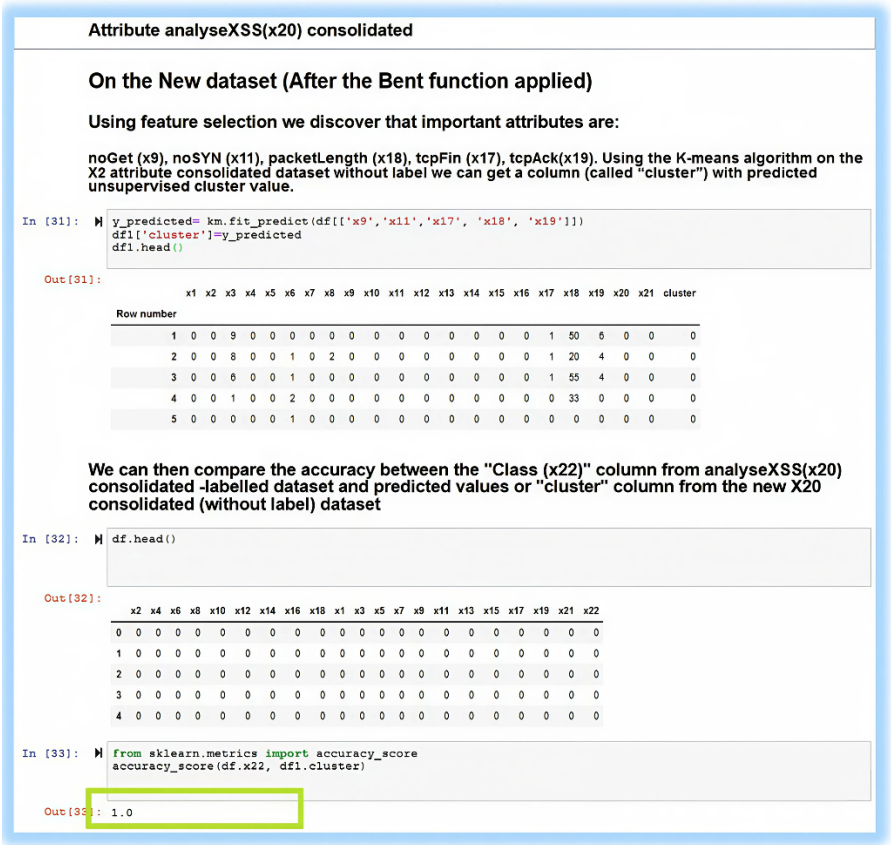

Figure 5. Univariate feature selection performance and visualization.

4.3. Execution of the Maiorana–McFarland Bent Function

The focus of the methodology was on executing the Maiorana–McFarland (M-M) Bent function defined as follows:

$$F(x, y) = x \cdot \pi(y) + h(y) \tag{6}$$

Key Elements:

- x: A single binary attribute selected from the network data.
- y: The remaining binary attributes in the network data.
- $\pi(y)$: A fixed permutation applied to the y attributes.
- $h(y)$: An arbitrary Boolean function of the y attributes.

Since the datasets contain only integer-valued data instead of binary inputs, a numerical mapping approach is used: the integer features are first converted into binary form, and then binary operations are performed using the Bent function. This allows the function to be applied within the classification framework to distinguish attack from non-attack records.

- **Transformation of Data**

The goal is to convert each feature into a single-bit value of 0 or 1 using the Bent function. After transforming the attributes into bits, they are mapped to either "Attack" or "Not Attack." A mod 2 operation is applied to the integer value to extract individual bits. A mod 2 binarization is a form of binary number representation that maps numbers to a single bit whose parity depends on the number. It deliberately ignores other numeric information and keeps a property unchanged despite changes in input size, such as varying packet counts in DDoS attacks. This pattern-focused method reduces noise and yields strong features, thereby improving the quality of the presented data and reducing overfitting to a greater extent than raw data. Finally, a logical decision rule is used to assign the final label. If at least two of the three selected attributes have a value of 1, the output is labeled 1 (Attack). If not, it is labeled 0 (Not Attack).

The majority-rule technique "two out of three bits equals one" was used to make the final attack decision accurately and reliably. By requiring that at least two independent groups agree on the attack decision, this method minimizes the impact of isolated errors and noise interference. Testing showed a clear distinction between attacks and typical behavior, a lower frequency of false alarms, and consistent performance in noisy environments. Therefore, this majority-rule technique represents a confident and effective means of making the final attack decision.

- **Permutation $\pi(y)$:**

In the permutation stage, the dataset's attributes are permuted in a consistent order. By consistent order, all even-numbered attributes are placed before the odd-numbered attributes, for example, x_2, x_4, ..., x_{20}, followed by x_1, x_3, ..., x_{21}. From this new set, one attribute is identified as the most valuable to the model through feature selection, denoted as x in the M-M equation. The remaining 20 attributes are labeled y and will be used for permutation and mapping.The permutation π is used to fix the even-odd order and is applied to every record in this data set. This method produces consistent, reliable results that are always easily identifiable. Tests have shown that using a random order yields inconsistent results, whereas a fixed order consistently yields reliable results. To ensure it can be repeated, Python performs this systematic reordering as follows:

```
# Fixed permutation for the whole data set
groupEven = [ 'x2', 'x4', 'x6', 'x8', 'x10','x12','x14', 'x16', 'x18', 'x20', ]
groupOdd = ['x1', 'x3', 'x5','x7','x9', 'x11', 'x13', 'x15', 'x17','x19','x21']
new_cols = groupEven + groupOdd
df.drop(['x22'], axis = 1, inplace = True)
df.columns
set(df.columns) == set(new_cols)
df0 = df[new_cols]
```

Column rearrangement is a key part of permutation visualization, as shown in Figure 6.

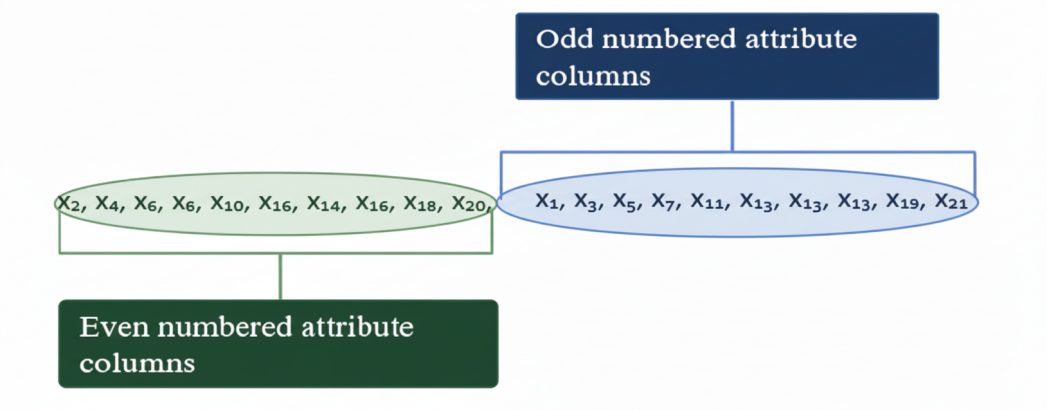

Figure 6. Fixed permutation applied to the entire dataset

- **Arbitrary Mapping (h(y))**

To address a Python library limitation that restricts permutation calculations to a maximum of 8 variables and the associated memory requirements, we divided our 20 attributes into 3 sets. Each set contained 7, 7, and 6 attributes, respectively (the first two sets contained the same number of attributes and could therefore be treated equally). The split was designed to provide a balanced view of all 20 attributes for our purposes without introducing any bias. By maintaining this 7-7-6 distribution, we have been able to generate permutations from these attributes accurately and effectively while still adhering to the strict scheduling and memory restrictions of the Bent function. For example:

Subgroup Computations:

$$h_1 = (x_1x_2 + x_3x_4 + x_5x_6)x_7 \tag{7}$$

$$h_2 = (x_8x_9 + x_{10}x_{11} + x_{12}x_{13})x_{14} \tag{8}$$

$$h_3 = (x_{15}x_{16}x_{17} + x_{18}x_{19}x_{20}) \tag{9}$$

Combined Output: Each subgroup produced an h output: h_1, h_2, and h_3. We can assign:

$$h = h_1 + h_2 + h_3 \tag{10}$$

- **Transformation:** This value h was then multiplied by an identity matrix $I_{1\times 20}$ and added to the M-M function.
- **Dataset Expansion:** This process creates 21 new datasets from each original dataset, each with 20 columns.
- **Mapping Insight:** In the figure below, we see how cost and balanced mappings can be applied to up to 7 variables, which are essential for constructing $h(y)$.

Its implementation in python is as follows:

 # Construction Bent function to apply to different rows
def BentFunction(i):
x = df0.iloc[i,0] # For the automation x = df00.iloc[i,0]
y = df00.iloc[i].values #y = df000.iloc[i].values
xpi = x * y
h1 = ((y[0] * y[1]) + (y[2] * y[3]) + (y[4] * y[5])) * y[6]
h2 = ((y[7] * y[8]) + (y[9] * y[10]) +(y[11] * y[12])) * y[13]
h3 = (y[14] * y[15] * y[16]) + (y[17] * y[18] * y[19])
h = h1 + h2 + h3
i = np.array([1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0])
hy = h*i
fxy = xpi + hy
return(fxy)

This value was then multiplied by an identity matrix ($I_{1\times 20}$) and added to the M-M function. It creates 21 new datasets from each original dataset, each with 20 columns. In the table below, we see how cost and balanced mappings can be applied to up to 7 variables, which are essential for constructing $h(y)$. Figure 7 demonstrates that by extending the coset representatives while maintaining balancedness, balanced function mappings can be successfully created for both six and seven variables.

Coset Representative f_i	Balanced Functions in f_i + R(2, 7)
0	300,503,590
$x_1x_2x_3$	171,425,760
$(x_1x_2 + x_3x_4)x_5$	156,664,832
$x_1x_2x_3 + x_4x_5x_6$ ⟵ Mapping with 6 variables	152,199,168
$(x_1x_2 + x_3x_4)x_5 + x_1x_3x_6$	153,664,128
$x_1x_2x_3 + x_4x_5x_6 + (x_1 + x_4)(x_2 + x_5)(x_3 + x_6)$	151,797,760
$(x_1x_2 + x_3x_4 + x_5x_6)x_7$ ⟵ Mapping with 7 variable	152,985,600
$(x_1x_2 + x_3x_4)x_5 + x_1x_6x_7$	151,417,600
$x_1x_2x_3 + x_4x_5x_6 + (x_1 + x_4)(x_2 + x_5)x_7$	151,169,024
$(x_1x_2 + x_3x_4)x_5 + x_1x_3x_6 + x_2x_3x_7$	152,182,784
$x_1x_2x_3 + x_4x_5x_6 + (x_1 + x_4)(x_2 + x_5)(x_3 + x_6) + x_1x_4x_7$	151,080,960
$x_1x_2x_3 + x_4x_5x_6 + x_2x_3 + x_4x_6 + ((x_1 + x_4)(x_2 + x_5) + x_3x_6))x_7$	151,057,408

Figure 7. Mapping of Balanced Function.

In Figure 8, the present diagram illustrates an xn consolidation, which is essential for DDoS and XSS-specific flow:

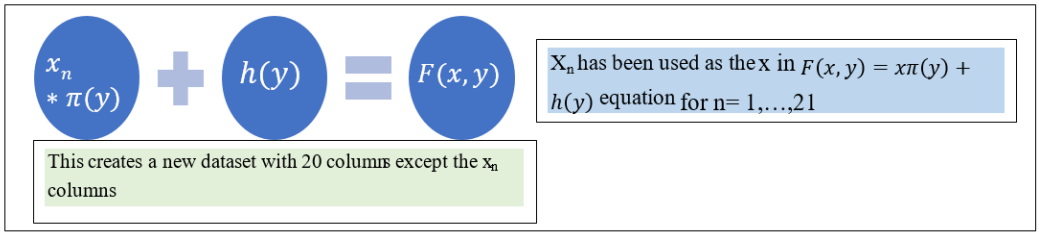

Figure 8. Function applied on D1–D4 for the DDoS and XSS attacks.

4.4. Machine Learning Evaluation of Bent Functions

The experimental assessment used both supervised and unsupervised machine learning techniques to evaluate the effectiveness of the Maiorana–McFarland (M-M) Bent function transformation. In the supervised machine learning part, Decision Tree algorithms were applied to labeled datasets (Dataset 1 for DDoS attacks, Dataset 3 for XSS attacks) because of their easy interpretation of numerical data, low data preparation needs, and robustness to outliers. For the unsupervised machine learning part, K-means clustering was

used on unlabeled data (Datasets 2 and 4) to form dense, distinguishable clusters. Model performance was measured using a 66–34% train/test split with 10-fold cross-validation in WEKA. The performance metrics included accuracy, Kappa statistics, mean absolute error, root mean squared error, and confusion matrices. The optimization process evaluated all 21 attributes individually through the M-M Bent function transformation.

Figure 9 shows the Decision Tree (J48 algorithm) for DDoS labeled data, which is the dataset before transformation. It indicates that the attributes `noGet` (x_9) and `dnsEnquiries` (x_4) were the main decision nodes used to predict DDoS attacks versus regular network traffic.

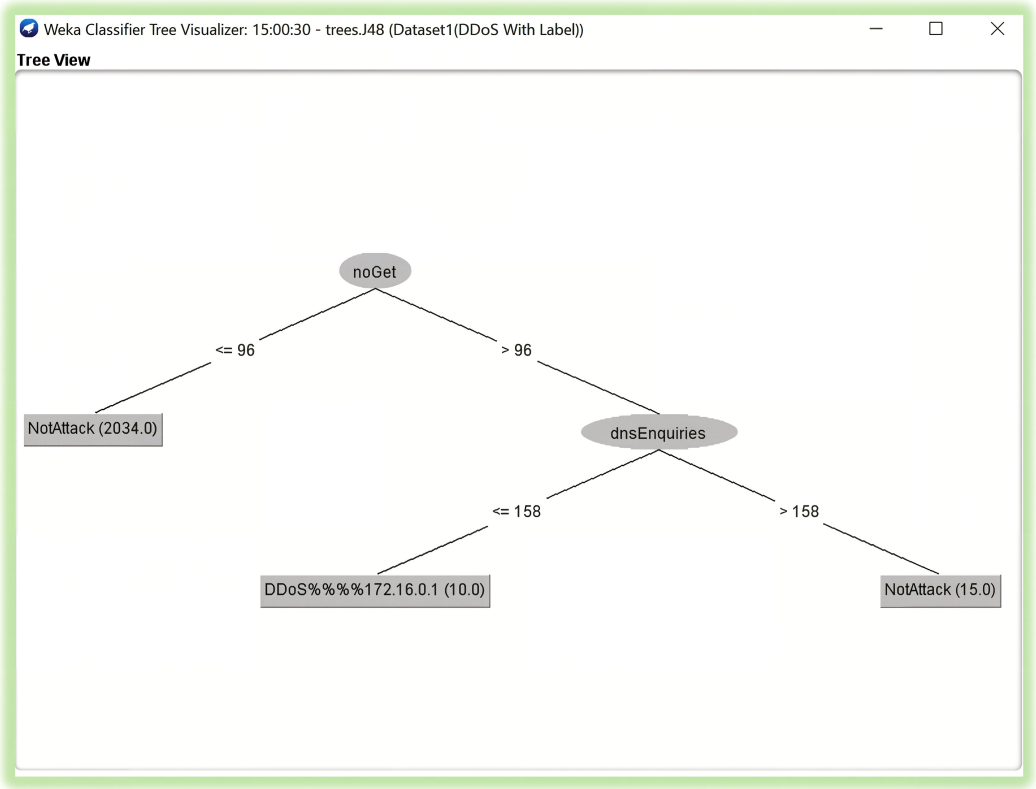

Figure 9. Decision Tree applied on DataSet1 (DDoS labeled) before the Bent function was applied.

4.4.1. Supervised Learning Results

The M-M Bent transformation consistently improves classification performance, as shown in the Supervised Learning Results table below. The two separate models were trained and tested on the dataset: a Decision Tree Classifier and a Random Forest Classifier. The results evidently demonstrate a significant improvement, at least in accuracy, after applying the transformation. Both models achieved perfect accuracy of 1.0 (100%) on the test data. This indicates that the M-M Bent transformation likely plays a crucial role in enhancing model performance on this dataset. However, it also raises concerns about potential data leakage or overfitting, making it important to further evaluate how well the models generalize to new, unseen data.

- **Accuracy of classification after applying the bent function:**

Now, to compare the results obtained after applying the bent function in both the DDoS and XSS cases and to provide a comprehensive view of the performance evaluation, the results are illustrated in Figures 10–13 below.

A. For the DDOS attack data—Decision Tree:

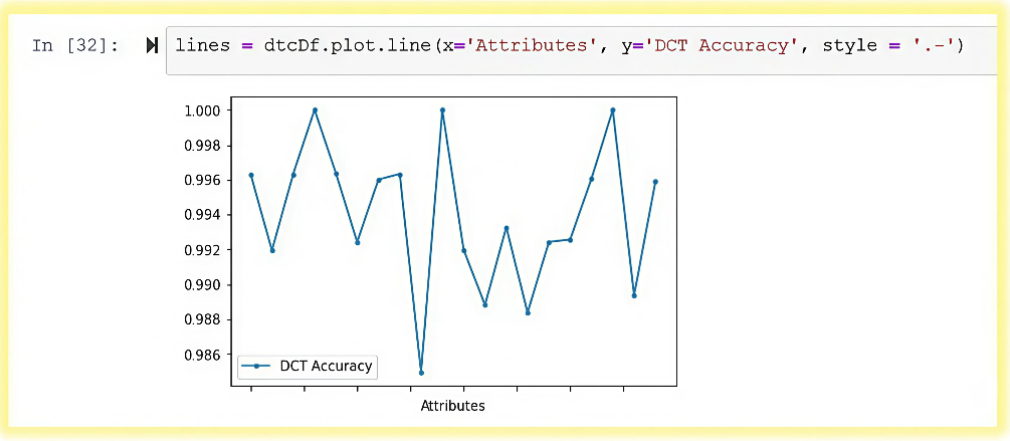

Figure 10. The Decision Tree – accuracy score for new DDoS attack data.

B. For the DDOS attack data—Random Forest:

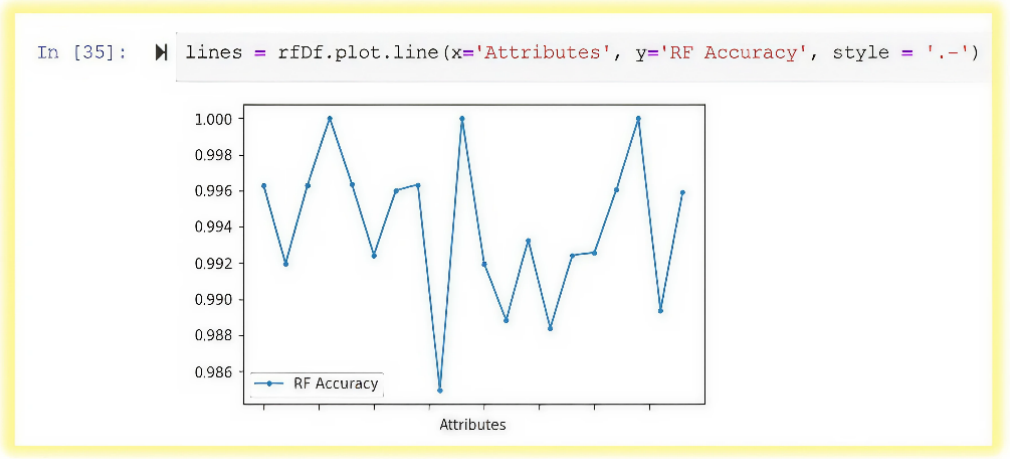

Figure 11. The Random Forest – accuracy score for new DDoS attack data.

C. For the XSS attack—Decision Tree:

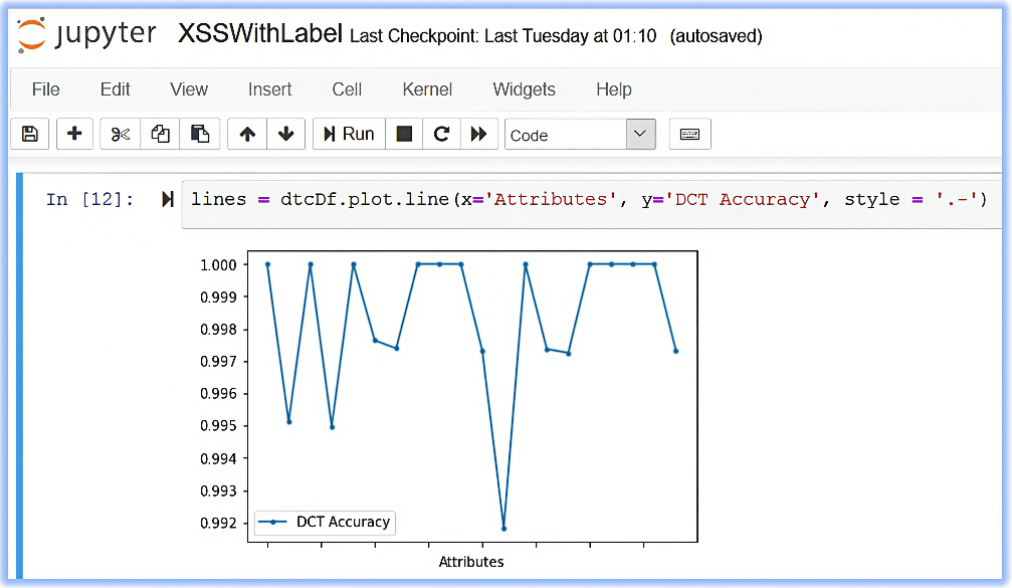

Figure 12. The Decision Tree accuracy score for XSS attributes.

D. For the XSS attack—Random Forest:

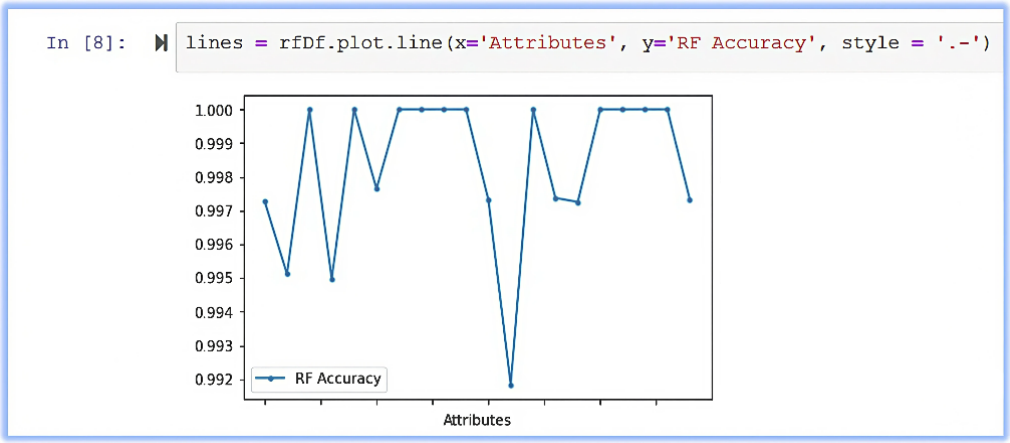

Figure 13. The Random Forest – accuracy scores for XSS attributes.

Table 3 shows the research results supporting the validity of the Maiorana–McFarland (M-M) Bent function as a preprocessing step in authentic learning environments. Although the initial image (supervised, classified) results demonstrated 100% accuracy (1.0) for both the Decision Tree and Random Forest classifiers with labeled data, the XSS data from Dataset 3 produced a critical score, indicating overfitting to the sample. Introducing the M-M Bent function increased data variability, helping to lessen this effect. The final accuracy score after applying the Random Forest model to the processed XSS dataset improved to 0.871891 (87.19%) using combinations of attributes X2 or X4.

According to Table 3, the Random Forest model achieved 100% accuracy when trained on the original feature set of an attribute; however, this result indicates that the Model was highly sensitive to the specific training data and did not exhibit stable predictive performance. After the Bent transformation, accuracy dropped significantly to 87.19% with the consolidated attributes, indicating reduced overfitting and more generalizable learning from the data. The modification will help reveal the true classification ability, as it becomes apparent under more realistic data conditions. This confirms that the Bent function is a valid component in authentic learning environments, improving data performance in detecting potentially malicious DDoS or XSS attacks.

Table 3. M–M Bent Transformation in Supervised Learning.

Attribute	Decision Tree (D1)	Random Forest (D3)
Dataset	DDoS With Label	XSS With Label
Accuracy (pre-Bent)	100% (1.0). Correctly classified 700 out of 700 instances.	100% (1.0). Perfect classification and overfitting data.
Accuracy (post-Bent)	Varies, up to 100% (1.0). Retains maximum accuracy when consolidated with X18 or X19 attributes.	87.19%. Consolidated attributes X2 or X4 achieved 0.871891.

4.4.2. Unsupervised Learning Results

Using the Bent function improves clustering performance in unsupervised learning. After applying the Bent function, the clusters became denser (more similar within each cluster) and more distinct from one another (greater differences between clusters). This indicates that the Bent function aids in data normalization and enhances the ability to detect attack-related data. As shown in measures such as the F1 score and Mean Squared Error (MSE) in Table 4, there is a noticeable increase in the accuracy of both classification and

clustering methods after using the Bent function. Cluster density, essential for unsupervised validation, is depicted in Figure 14.

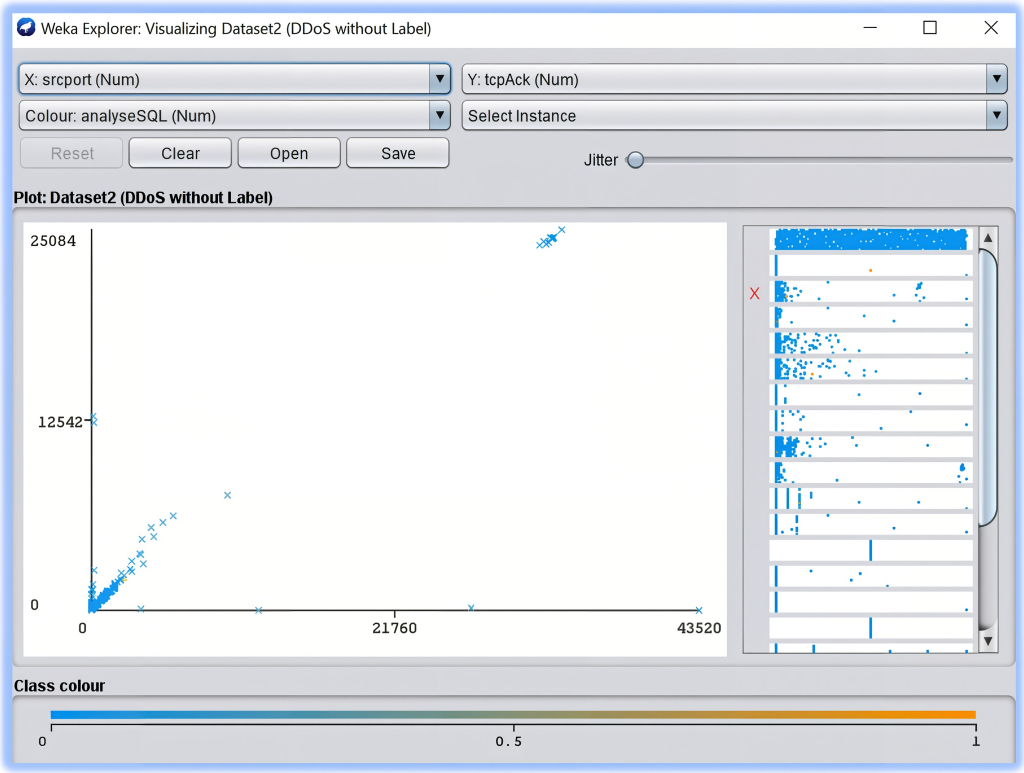

Figure 14. K-Means clustering on Dataset 2 (DDoS without label) before the Bent function was applied.

Table 4 compares the performance of unsupervised learning methods, including K-Means and Hierarchical Clustering, applied to the unlabelled DDoS (D2) and unlabelled XSS (D4) datasets. The main metrics evaluated were Accuracy, F1 Score, and Mean Squared Error (MSE). The results were analyzed before and after transformation using the Maiorana–McFarland (M-M) Bent function. Although D2 and D4 are unlabelled, metrics for clustering, such as F1 Score and MSE, often require linking them to the corresponding labelled datasets (D1 and D3). In particular, accuracy was measured by mapping the generated cluster labels to the ground-truth class labels of these parallel datasets. It should be noted that the difference between the pre- and post-Bent measures (e.g., 99.85% vs. 99.92) is small and does not constitute substantive improvement. Instead, the preservation of outstanding performance is the most significant outcome. The fact that this cryptographic transformation can be performed while maintaining a high clustering algorithm quality demonstrates the approach's compatibility. It also proves its strength in retaining the internal data structure. Finally, the figures in the New Data columns present the optimized score across the 21 datasets generated by applying the Bent function.

Table 4. M–M Bent Transformation in Unsupervised Learning.

Algorithm	Dataset	Metric	Original Data (Pre-Bent)	New Data (Post-Bent)
K-Means (K = 2)	D2 (DDoS unlabelled)	Accuracy	99.85%	99.92% (Observed for important features)
K-Means (K = 2)	D2 (DDoS unlabelled)	F1 Score	0.9989	0.9991
K-Means (K = 2)	D2 (DDoS unlabelled)	MSE	0.0003	0.0000

Table 4. *Cont.*

Algorithm	Dataset	Metric	Original Data (Pre-Bent)	New Data (Post-Bent)
K-Means (K = 2)	D4 (XSS unlabelled)	Accuracy	98.94% (Observed initial accuracy)	100% (Observed optimized accuracy)
Hierarchical (Agglomerative)	D4 (XSS unlabelled)	Accuracy	99.84%	N/A
Hierarchical (Agglomerative)	D4 (XSS unlabelled)	F1 Score	0.9969	0.9969
Hierarchical (Agglomerative)	D4 (XSS unlabelled)	MSE	0.0000	0.0016

4.5. Discussion

The findings of this study clearly indicate that using Bent functions, especially within the Maiorana–McFarland construction, offers a new and effective method for preprocessing data and improving machine learning performance in both supervised and unsupervised settings. The unique mathematical qualities of Bent functions, particularly their balancedness and nonlinearity, help transform datasets to generate more distinct feature spaces. This results in better clustering outcomes. The improved performance in unsupervised learning is evident here, particularly for network-based datasets used in DDoS and XSS attack detection. The results with Bent functions demonstrate the benefit of applying cryptographic concepts to data-driven machine learning models to strengthen anomaly detection and pattern recognition in cybersecurity.

Using Bent functions as an initial preprocessing step in the machine learning pipeline provides a significantly different perspective than the mostly statistical or neural feature extraction used in current research. This paper presents a transformation from cryptography to a preprocessing method utilizing Bent functions. This approach helps machine learning algorithms identify critical attributes without introducing bias into classifiers trained on imbalanced data. This research integrates cryptographic theory and machine learning by repurposing Bent functions for feature engineering, thereby bridging the conceptual gap between these domains. Finally, applying this approach to three datasets involving the detection of real and simulated cyberattacks demonstrates the value of an algorithm that captures the complexities of diagnosing and recognizing subtle data pattern changes associated with anomalous previous patterns, which could be considered faults requiring interception or intervention.

The experiment's reliability was confirmed using various analytical tools, including Python's Jupyter Notebook and the WEKA platform. These tools enabled algorithmic verification and allowed for monitoring of GUI-based systems, aiding in cross-experiment validation. However, some computational and technical limitations were encountered. The permutation implementation of the Maiorana–McFarland Bent function relied on Python's built-in library, which restricted the number of variables to 8. This limitation prevented applying the function to datasets with more than 20 attributes. Additionally, manually creating datasets in Excel for each experiment created scalability and automation challenges, highlighting the need for more advanced data generation and management methods in future research.

While WEKA's graphical interface helped visualize results, the lack of built-in functionality for applying the Bent function algorithm likely limited the scope of experiments. In the future, it may be beneficial to explore compatibility with the Bent function in WEKA's Java API or other machine learning libraries. This would facilitate smoother automated testing, enhance reproducibility, and reduce the likelihood of programming errors. Additionally, time constraints prevented this study from investigating alternative Bent function constructions and their relative effectiveness. Expanding the range of such constructions and

examining increased randomness in the permutation could further clarify how different mathematical approaches can enhance sampling-based learning.

A methodological limitation arises from applying the Bent function transformation to all datasets before cross-validation. As it currently stands, during cross-validation, information from the test folds could have leaked into the training folds because the data were transformed before splitting into separate groups. Because this design was used to evaluate the overall effect of the transformation, a more stringent design would have fit the transformation parameters separately for each training fold within each cross-validation round. An additional point of importance is that model performance was not derived from a single data split; instead, a 10-fold cross-validation scheme was used to improve stability and reduce partition bias. For future work, we will perform fold-wise transformations, use multiple random initialization procedures, and evaluate the resulting model's generalization on an external test set to ensure unbiased and reliable assessment.

To summarize, the research shows that Bent functions enhance clustering accuracy and feature separation when used in complex cybersecurity datasets. Although there are some computational and methodological limitations, this study confirms the significance of cryptographically derived functions in improving learning algorithms. Further research into combining Bent function theory with scalable machine learning systems could boost the accuracy and robustness of cyber threat detection systems.

5. Conclusions and Future Work

This research has demonstrated the effectiveness of using the Bent function as a learning tool in ML frameworks for cyber-attack detection. Experiments applying the Bent function to existing DDoS and XSS datasets showed significant improvements in clustering performance and classification accuracy. These improvements were reflected in key performance metrics, including the F1 score, Mean Squared Error (MSE), and Confusion Matrix metrics. The study reveals that the Bent function, known for its maximum nonlinearity and minimal autocorrelation, can enhance data representation and boost learning performance in both supervised and unsupervised architectures. Furthermore, by using the Maiorana–McFarland construction and fixed permutations, a consistent level of entropy was maintained, simulating a realistic, variable data environment. This approach improves feature separation and pattern recognition in cyberattack datasets.

Although the study did not definitively identify which of the 21 derived datasets delivered the best predictive performance, the experimental results clearly demonstrate that the Bent function can be used in pre-processing or feature transformation to improve the generalizability and effectiveness of detection models while mitigating overfitting. This research also provides a conceptual foundation for applying cryptographic principles, including those related to the Bent function, to cybersecurity analytics. This approach will help enhance efficiency and accuracy in threat detection. Several areas for future work are suggested. First, exploring different constructions of the Bent function will help pinpoint specific structural modifications that boost data optimization and classification results. Future experiments could incorporate more advanced or diverse machine learning algorithms, such as ensemble models, deep neural networks, or hybrid systems, to compare their performance with methods applied to Bent-transformed datasets. Automating implementation through APIs or data pipelines will also ensure broader deployment. Consistency in this process will support analysis of larger datasets and real-world data.

Building on the foundation established by this study, subsequent research will develop a more detailed and thorough evaluation framework. Specifically, we will include (a) complete cross-validation with a confidence interval-based assessment of statistical significance; (b) explicit checking of the data transformation process to ensure the absence of data

leakage by limiting the fitting of transformation processes to a small portion of the data after the data have been split into training and validation data sets; (c) comprehensive studies to isolate the effects of specific properties and configurations of the Bent functions; and (d) a variety of comparisons with additional sets of benchmark models. These steps must be taken to provide complete validation of the strength, practicality, and generalizability of our method.

In conclusion, this study provides a strong foundation for linking cryptographic theory to machine learning. Additionally, it shows that Bent functions are a promising method for innovative data transformation to enhance cyberspace attack detection and adaptive learning environments.

Author Contributions: Conceptualization: N.K.; methodology: N.K.; validation: S.M.K. and N.K.; formal analysis: S.M.K. and N.K.; investigation: S.M.K. and N.K.; resources: S.M.K. and N.K.; writing—original draft preparation: S.M.K. and N.K.; writing-review and editing: S.M.K. and N.K.; visualization: S.M.K. and N.K.; supervision: S.M.K. and N.K.; project administration: S.M.K. and N.K.; funding acquisition: S.M.K. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by School of Science and Technology, City St George's, University of London— Invoice Number: 4055595.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Cross-Site Scripting (XSS) dataset for deep learning, along with the CICDDoS2019 dataset available on Kaggle [https://www.kaggle.com/datasets/dhoogla/cicddos2019], (accessed on 4 January 2026).

Acknowledgments: We would like to thank the School of Science and Technology, City St. George's University of London, for supporting this research and Shahnewaz Adnan Khan for undertaking the experiment as part of his MSc thesis.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sriranjani, R.; Saleem, M.D.; Hemavathi, N.; Parvathy, A. A Machine Learning Based Intrusion Detection Scheme to Detect Replay Attacks in the Smart Grid. In Proceedings of the 2023 IEEE International Students' Conference on Electrical, Electronics, and Computer Science (SCEECS), Bhopal, India, 18–19 February 2023; pp. 1–5. https://doi.org/10.1109/SCEECS57921.2023.10063021.
2. Liu, W.; Zhao, Q.; Zhang, Q. Construction of Vectorial Boolean Functions with Provable Differential-Linear Uniformity and High Nonlinearity. In Proceedings of the 2024 6th International Conference on Natural Language Processing (ICNLP), Hangzhou, China, 22–24 March 2024; pp. 300–304. Available online: https://ieeexplore.ieee.org/document/10692619 (accessed on 1 November 2025).
3. Ni, T.; Lan, G.; Wang, J.; Zhao, Q.; Xu, W. Eavesdropping Mobile App Activity via Radio-Frequency Energy Harvesting. In Proceedings of the 32nd USENIX Security Symposium 2023, Anaheim, CA, USA, 9–11 August 2023; pp. 3511–3528. Available online: https://www.usenix.org/conference/usenixsecurity23/presentation/ni (accessed on 20 December 2025).
4. Li, J.; Wu, S.; Zhou, H.; Luo, X.; Wang, T.; Liu, Y.; Ma, X. Packet-Level Open-World App Fingerprinting on Wireless Traffic. In Proceedings of the 2022 Network and Distributed System Security Symposium (NDSS'22), San Diego, CA, USA, 24–28 April 2022; pp. 1–12. Available online: https://www.ndss-symposium.org/ndss-paper/auto-draft-218/ (accessed on 21 December 2025).
5. Li, J.; Zhou, H.; Wu, S.; Luo, X.; Wang, T.; Zhan, X.; Ma, X. FOAP: Fine-Grained Open-World Android App Fingerprinting. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 1579–1596. Available online: https://www.usenix.org/conference/usenixsecurity22/presentation/li-jianfeng (accessed on 22 December 2025).
6. Kumar, J.H.; Ponsam, J.G. Cross Site Scripting (XSS) Vulnerability Detection Using Machine Learning and Statistical Analysis. In Proceedings of the 2023 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 23–25 January 2023; pp. 1–9. https://doi.org/10.1109/ICCCI56745.2023.10128470.

7. Nilavarasan, G.; Balachander, T. XSS Attack Detection Using Convolution Neural Network. In Proceedings of the 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 5–7 January 2023; pp. 1–6. https://doi.org/10.1109/ICECONF57129.2023.10083807.
8. Lu, D.; Liu, L. Research on Cross-site Scripting Attack Detection Technology Based on Few-shot Learning. In Proceedings of the 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 24–26 February 2023; pp. 1425–1429. https://doi.org/10.1109/ITNEC56291.2023.10082596.
9. Sethi, M.; Verma, J.; Snehi, M.; Baggan, V.; Virender.; Chhabra, G. Web Server Security Solution for Detecting Cross-site Scripting Attacks in Real-time Using Deep Learning. In Proceedings of the 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1), Noida, India, 21–22 April 2023; pp. 1–5. https://doi.org/10.1109/ICAIA57370.2023.10169255.
10. Hamzah, K.H.; Osman, M.Z.; Anthony, T.; Ismail, M.A.; Abdullah, Z.; Alanda, A. Comparative Analysis of Machine Learning Algorithms for Cross-Site Scripting (XSS) Attack Detection. *JOIV Int. J. Inform. Vis.* **2024**, *8*, 1678–1685. https://joiv.org/index.php/joiv/article/view/3451.
11. Oshoiribhor, E.O.; John-Otumu, A.M. XSS-Net: An Intelligent Machine Learning Model for Detecting Cross-Site Scripting (XSS) Attack in Web Application. *Mach. Learn. Res.* **2025**, *10*, 14–24. https://doi.org/10.11648/j.mlr.20251001.12.
12. Alhamyani, R.; Alshammari, M. Machine Learning-Driven Detection of Cross-Site Scripting Attacks. *Information* **2024**, *15*, 420. https://doi.org/10.3390/info15070420.
13. Bakır, R.; Bakır, H. Swift Detection of XSS Attacks: Enhancing XSS Attack Detection by Leveraging Hybrid Semantic Embeddings and AI Techniques. *Arab. J. Sci. Eng.* **2025**, *50*, 1191–1207. https://doi.org/10.1007/s13369-024-09140-0.
14. Ahmim, A.; Maazouzi, F.; Ahmim, M.; Namane, S.; Dhaou, I.B. Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model. *IEEE Access* **2023**, *11*, 119862–119875. https://doi.org/10.1109/ACCESS.2023.3327620.
15. Prathap, P.; Duttagupta, S. AI-Enabled Fast Detection of DDoS and Adversary DDoS Attacks in SDN. In Proceedings of the 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), IIT Delhi, New Delhi, India, 6–8 July 2023; pp. 1–7. https://doi.org/10.1109/ICCCNT56998.2023.10306608.
16. Kamaldeep.; Malik, M.; Dutta, M. Feature Engineering and Machine Learning Framework for DDoS Attack Detection in the Standardized Internet of Things. *IEEE Internet Things J.* **2023**, *10*, 8658–8669. https://doi.org/10.1109/JIOT.2023.3245153.
17. Haque, S.; El-Moussa, F.; Komninos, N.; Muttukrishnan, R. Identification of Important Features at Different IoT Layers for Dynamic Attack Detection. In Proceedings of the 2023 IEEE 9th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), Laguna Hills, CA, USA, 6–8 May 2023; pp. 84–90. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS58521.2023.00025.
18. Dantas, C.; Pessoa, P.; Ferreira, J.; Maciel, P.; Dantas, J. DDoS Detection Based on Hardware Performance Counters Selection. In Proceedings of the 2023 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), Biarritz, France, 14–16 December 2023; pp. 1–6. https://doi.org/10.1109/WETICE57085.2023.10501947.
19. Lamptey, R.; Saedi, M.; Stankovic, V. Machine-Learning Anomaly Detection for Early Identification of DDoS in Smart Home IoT Devices. In Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR), Venice, Italy, 4–6 August 2025; pp. 105–110. https://doi.org/10.1109/CSR64739.2025.11130003.
20. Kachavimath, A.V.; Narayan, D. An Efficient DDoS Attack Detection in SDN Using Multi-Feature Selection and Ensemble Learning. *Procedia Comput. Sci.* **2025**, *252*, 241–250. https://doi.org/10.1016/j.procs.2024.12.026.
21. Alnajim, A.M.; Alotaibi, F.M.; Khan, S. Detecting and Mitigating Distributed Denial of Service Attacks in Software-Defined Networking. *Comput. Mater. Contin.* **2025**, *83*, 4515–4535. https://doi.org/10.32604/cmc.2025.063139.
22. Xu, Z. Deep Learning Based DDoS Attack Detection. *ITM Web Conf.* **2025**, *70*, 03005. https://doi.org/10.1051/itmconf/20257003005.
23. Li, Y.; Kan, H.; Mesnager, S.; Peng, J.; Zheng, L. Direct Approaches for Generic Constructions of Plateaued Functions and Bent Functions Outside M#. *IEEE Trans. Inf. Theory* **2025**, *71*, 1400–1418. https://doi.org/10.1109/TIT.2024.3497804.
24. Kudin, S.; Pasalic, E.; Polujan, A.; Zhang, F.; Zhao, H. Almost Maiorana-McFarland Bent Functions. *IEEE Trans. Inf. Theory* **2025**, *71*, 9698–9713. https://doi.org/10.1109/TIT.2025.3614379.
25. Anbar, N.; Kudin, S.; Meidl, W.; Pasalic, E.; Polujan, A. Vectorial Negabent Concepts: Similarities, Differences, and Generalizations. *Des. Codes Cryptogr.* **2025**, *93*, 899–921. https://doi.org/10.1007/s10623-024-01454-2.
26. Kim, S.; Shin, M.; Kim, S.; Shin, H.; Kim, I.; Kwon, D.; Lee, D.; Kim, S.; Hong, D.; Sung, J.; et al. Redefining Security in Shadow Cipher for IoT Nodes: New Full-Round Practical Distinguisher and the Infeasibility of Key-Recovery Attacks. *IEEE Internet Things J.* **2025**, *12*, 6805–6817. https://doi.org/10.1109/JIOT.2024.3491138.
27. Çeşmelioğlu, A.; Meidl, W. Construction and Equivalence for Generalized Boolean Functions. *Cryptogr. Commun.* **2025**, *17*, 1659–1682. https://doi.org/10.1007/s12095-025-00805-7.
28. Li, K.; Li, C.; Helleseth, T.; Qu, L. Further Investigations on Permutation Based Constructions of Bent Functions. *J. Comb. Theory, Ser. A* **2023**, *199*, 105779. https://doi.org/10.1016/j.jcta.2023.105779.

29. Cardell, S.D.; Fuúter-Sabater, A.; Requena, V.; Beltrá, M. A New Representation of Binary Sequences by Means of Boolean Functions. *arXiv* **2025**, arXiv:2506.05374. https://doi.org/10.48550/arXiv.2506.05374.
30. Prévost, T.; Martin, B. Testing a Cellular Automata Construction Method to Obtain 9-Variable Cryptographic Boolean Functions. *arXiv* **2025**, arXiv:2501.09380. https://doi.org/10.48550/arXiv.2501.09380.
31. Pasalic, E.; Polujan, A.; Kudin, S.; Zhang, F. Design and Analysis of Bent Functions Using M-Subspaces. *IEEE Trans. Inf. Theory* **2024**, *70*, 4464–4477. https://doi.org/10.1109/TIT.2024.3352824.
32. Polujan, A.; Mariot, L.; Picek, S. On Two Open Problems on the Normality of Bent Functions. *Discret. Appl. Math.* **2025**, *360*, 115–118. https://doi.org/10.1016/j.dam.2024.08.023.
33. Polujan, A.; Pasalic, E.; Kudin, S.; Zhang, F. Bent Functions Satisfying the Dual Bent Condition and Permutations with the (A_m) Property. *Cryptogr. Commun.* **2024**, *16*, 1235–1256. https://doi.org/10.1007/s12095-024-00724-z.