sustainability

Article

Energy Analysis-Based Cyber Attack Detection by IoT with Artificial Intelligence in a Sustainable Smart City

D. Prabakar [1], M. Sundarrajan [2], R. Manikandan [3], N. Z. Jhanjhi [4,*], Mehedi Masud [5] and Abdulmajeed Alqhatani [6]

[1] Department of Data Science and Business Systems, School of Computing, SRM Institute of Science and Technology, Kattankulathur Campus, Chennai 603203, India
[2] Department of CSE, SRM Institute of Science and Technology, Ramapuram Campus, Chennai 600089, India
[3] School of Computing, SASTRA Deemed University, Thanjavur 613401, India
[4] School of Computer Science, SCS, Taylor's University, Subang Jaya 47500, Malaysia
[5] Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia
[6] Department of Information Systems, College of Computer Science & Information Systems, Najran University, Najran 61441, Saudi Arabia
* Correspondence: noorzaman.jhanjhi@taylors.edu.my

Abstract: Cybersecurity continues to be a major issue for all industries engaged in digital activity given the cyclical surge in security incidents. Since more Internet of Things (IoT) devices are being used in homes, offices, transportation, healthcare, and other venues, malicious attacks are happening more frequently. Since distance between IoT as well as fog devices is closer than distance between IoT devices as well as the cloud, attacks can be quickly detected by integrating fog computing into IoT. Due to the vast amount of data produced by IoT devices, ML is commonly employed for attack detection. This research proposes novel technique in cybersecurity-based network traffic analysis and malicious attack detection using IoT artificial intelligence techniques for a sustainable smart city. A traffic analysis has been carried out using a kernel quadratic vector discriminant machine which enhances the data transmission by reducing network traffic. This enhances energy efficiency with reduced traffic. Then, the malicious attack detection is carried out using adversarial Bayesian belief networks. The experimental analysis has been carried out in terms of throughput, data traffic analysis, end-end delay, packet delivery ratio, energy efficiency, and QoS. The proposed technique attained a throughput of 98%, data traffic analysis of 74%, end-end delay of 45%, packet delivery ratio of 92%, energy efficiency of 92%, and QoS of 79%.

Keywords: cyber-attack; security; IoT devices; network traffic analysis; malicious attack detection; artificial intelligence; sustainable smart city

Citation: Prabakar, D.; Sundarrajan, M.; Manikandan, R.; Jhanjhi, N.Z.; Masud, M.; Alqhatani, A. Energy Analysis-Based Cyber Attack Detection by IoT with Artificial Intelligence in a Sustainable Smart City. *Sustainability* **2023**, *15*, 6031. https://doi.org/10.3390/su15076031

Academic Editor: Andreas Ihle

Received: 5 February 2023
Revised: 25 February 2023
Accepted: 7 March 2023
Published: 30 March 2023

1. Introduction

As physical systems become more interconnected with the internet, they become vulnerable to cyber-attacks. More than 30 surveys on the cybersecurity issue in CPSs were published, according to [1] published in 2017. With the rise of automated assaulting tools and the increased sophistication of cyber-attacks, professional hacking groups have begun to participate. Successful cyber-attacks could have disastrous, catastrophic, or even lethal results on a CPS [2]. However, protecting CPSs from cyber-attacks is difficult. The lack of cybersecurity features such as message authentication in many CPS systems makes it difficult to determine fraudulent data injection attacks. It is difficult to protect against eavesdropping assaults due to a lack of universal encryption, especially on systems using antiquated technologies. To stop replay assaults, it is necessary to refer to system states. Additionally, the majority of the time, an outdated method used in operation restricts options for network traffic protection. Considering how the Internet of Things affects our

daily lives and how swiftly its application areas are growing, it is most likely the greatest modern invention [3].Deep learning (DL) outperforms conventional machine learning (ML) solutions in terms of performance. When there is enough information, DL methods nearly always produce great results. In contrast to other domains such as NLP, image processing, software vulnerability, and many more [4], DL methods have just recently been used to address the CPS cybersecurity issue. Additionally, it has been noted that a large number of DL models have been suggested in recent articles to identify CPS cyber-attacks. The degree of complexity when superimposing cybersecurity over CPSs was attributed as a widely recognised explanation for why it is difficult to detect cyber-attacks on CPSs [5]. ML methods are utilised in tasks such as regression as well as classification because they have the capacity to infer useful knowledge from data produced by humans or machines. Similarly, ML can be applied to offer security services in an IoT network. ML is being employed more and more in many applications in the cybersecurity industry and its usage in attack detection difficulties is becoming a fiercely debated topic [6].

The contribution of this research is as follows:

1. To propose a novel method in cybersecurity-based network traffic analysis and malicious attack detection using IoT artificial intelligence techniques for a sustainable smart city;
2. The traffic analysis has been carried out using a kernel quadratic vector discriminant machine which enhances the data transmission by reducing network traffic;
3. The malicious attack detection is carried out using adversarial Bayesian belief networks.

The organization of this article is as follows: Section 2 gives existing technique based on network traffic and attack detection, Section 3 gives proposed research and its experimental analysis has been carried out in Section 4. The Section 5 concludes research with future scope.

2. Related Works

There are a few interesting deep learning-based research projects in the cybersecurity field, despite the fact that deep learning research has currently prospered in fields such as pattern recognition, image processing, and text processing. The earlier works of [7] demonstrate that DLNN, either as a standalone method or in combination with optimization or ML methods [8], can predict assaults with great accuracy. More specifically, [9] integrate SVMs with ANNs, which dramatically improve detection rates over standalone DL or ML techniques. In particular, [10] develops hybridization by fusing SVM and ANN, adding a genetic algorithm (GA) and PSO to that fusion. A 99.3% accuracy rate is achieved by this hybridization. The man shift technique was tested by [11] using the KDD99 network traffic dataset to identify network invasion. The mean shift could, according to the authors, identify an assault in the network dataset. However, user to root (U2R) and remote to local (R2L) assaults were not picked up by the algorithm.

Serra and others offer a new method for adaptive clustering utilizing GANS, by [12] introduced ClusterGAN. A network intrusion detection system (NIDS) was created by Choi et al. using unsupervised learning versus unlabeled data. To identify FDI (False Data Injection) assaults, work [13] assessed SVM, KNN, and ANN. According to the findings of their trial, KNN and SVM were more accurate than ANN. A function that maps an input to an output is learned through supervised learning using examples (labelled data) of such input-output pairs. By using two open-source NIDS as well as two supervised ML approaches on backscatter darknet traffic, [14–16] examined the effectiveness of various supervised ML methods in recognizing cyber-attacks, notably SYN-DOS attacks on IoT methods. The development of wireless sensor networks (WSN), correspondence innovation, and IoT innovation was documented by the authors in [17].

IDS-applicable ML methods such KNN, SVM, DT, NB, NN, and RF were used by the authors of [18]. On the Bot-IoT data collection, the authors compared ML methods for multi- and binary-class combinations. These models were utilized to determine the F1 score, recall, precision, and accuracy. In [19,20], which compares ML with deep-learning neural networks using an online dataset, the identification of assaults in FOG design is investigated. One of the famous location frameworks, Grunt [21], is likewise a mark-based

framework and utilizations assault signature rules to recognize the digital assaults. They utilize an example search calculation, called AhoCorasick [22], to conclude the approaching traffic design as assaults or not. Another location framework, Suricata [23], is a famous public IDS, completely upholds multithreading engineering, and is more reasonable for enormous scope network frameworks. The review utilized the Suricata to carry out the discovery framework on the asset limitation gadget, Raspberry Pi. They expect to recognize the port checking assault on the IoT climate. Different investigations [24] likewise proposed the assault discovery framework for the IoT climate, and they zeroed in on port checking, MITM, DNS store harming, and flood assaults. The review [25] referenced that Grunt is lighter than Suricata. They likewise proposed the AI-based discovery structure to expand the Grunt framework. Their outcomes showed that the recognition consequences of their expansion are superior to the first Grunt. Table 1 shows comparison of energy analysis with cyber-attack detection.

Table 1. Comparison of existing technique based on energy analysis with cyber-attack detection.

Author	Description	Dataset	ML Algorithm
Work [8]	With the BoT IoT identification dataset being used, a novel framework model and a hybrid algorithm have been presented to address the difficulty of ML algorithms for cyber attacks.	BoT_IoT dataset	NB, bayesNEt, DT, RF
Work [9]	This paper suggests two semi-distributed and distributed approaches that combine high performance feature extraction and selection with potential fog-edge coordinated analytics to solve the drawbacks of centralised IDS for resource-constrained devices.	AWID dataset	SVM
Work [10]	Present an intelligent architecture that combines CEP and machine learning (ML) to quickly and accurately identify various IoT security breaches. In particular, such an architecture may easily manage event patterns whose criteria depend on values obtained by ML algorithms.	MQTT regular traffic packets	SVR
[11]	Using both datasets and actual network scenarios, this study examines how well DAS CIDS performs in the detection and false alarm reduction categories.	KDD 99	KNN, SVM, RF, DT
[12]	In order to identify and classify malware, IoT applications' opcodes are converted into a vector space and fuzzy and quick fuzzy pattern tree methods are used.	IoT, Vx-heaven, Kaggle and ransomware	FPT
[14]	Offers a new ELM-based ESFCM technique as well as assault detection based on fog.	NSL_KDD	Fuzzy C-means algorithm
[15]	Proposes a machine learning (ML) based attack detection model that can be trained on data and logs obtained by PMUs for use in power systems.	ICS cyber-attack datasets	KNN, SVM, DT, RF, XG boost
[16]	Using several ML techniques, anomaly and attack detection in IoT sensor data was compared.	Kaggle, message queuing telemetry transport (MQTT) protocol	LR, SVM, DT, RF
[17]	The authors suggest a network-centric, behavior-based anomaly detection approach for safeguarding IoT environments, where predictability of TCP traffic from IoT devices may be leveraged to quickly identify different DDoS attacks using unsupervised machine learning.	IoT traffic	SVM

3. System Model

This section discusses a novel technique in cybersecurity-based network traffic analysis and malicious attack detection using IoT artificial intelligence techniques for a sustainable smart city. The traffic analysis has been carried out using a kernel quadratic vector discriminant machine which enhances the data transmission by reducing network traffic. This enhances

the energy efficiency with reduced traffic. Then, the malicious attack detection is carried out using adversarial Bayesian belief networks. The proposed model is shown in Figure 1.

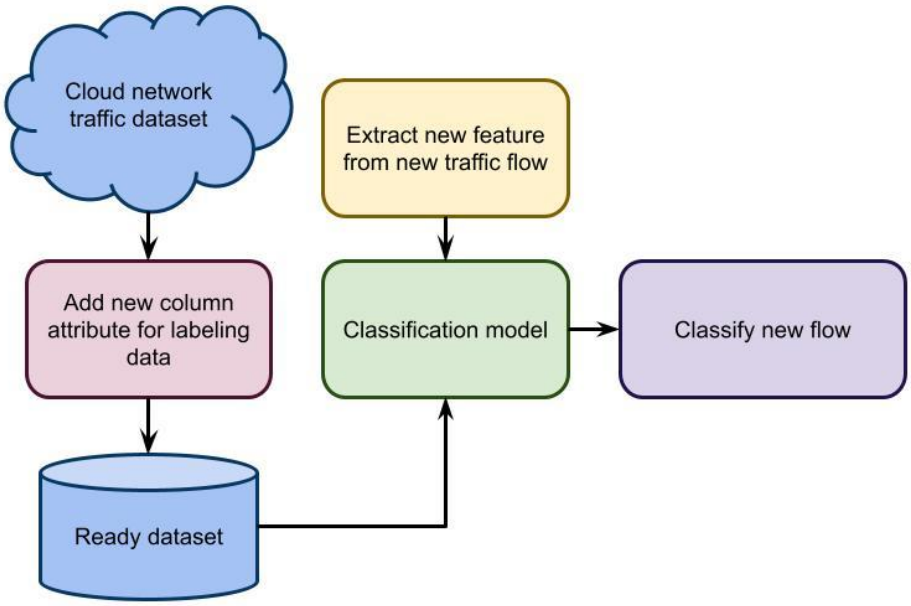

Figure 1. Proposed architecture of network traffic analysis and malicious attack detection.

Pre-processing data transformation techniques are utilized to transform a dataset into an ML-friendly structure. This step of cleaning the dataset also makes it more effective by getting rid of bad or unnecessary data that could make the accuracy of the dataset worse.

3.1. Kernel Quadratic Vector Discriminant Machine Based Traffic Analysis

Finding a separation surface to accurately separate two classes of data from a given dataset is the aim of binary classification. Data collection with two classes is mathematically denoted for any binary classification issue by Equation (1).

$$\mathcal{D} = \left\{ \left(x^{(i)}, y^{(i)} \right)_{i=1,\cdots,N} \mid x^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{-1, 1\} \right\} \tag{1}$$

Noting that N = N+ + N, denote their respective cardinalities as N+ and N. We assume that M+ and M are both nonempty in this article. To truly segregate the data using a classifier is the aim of binary classification. If u$\in\mathbb{R}^n$ and d$\in$R exist and are such that a dataset D can be linearly separated, then by Equation (2).

$$u^T x^{(i)} + d > 0 \left(i \in \mathcal{M}^+ \right), u^T x^{(i)} + d < 0, \left(i \in \mathcal{M}^- \right) \tag{2}$$

The goal of SVM is to maximise the margin of separation when separating a given linearly separable dataset D by a hyperplane. If you use the notation f(x) = $u^T x^d$x + d for separation function, the width of the margin is equal to $\frac{2}{\|u\|_2}$. The soft-margin idea is used if dataset D is not linearly separable by introducing slack vector $\xi = [\xi_1,...,\xi_N] T \in R_N$ to permit placement of points to violate constraints by Equation (3).

$$\min \frac{1}{2} \parallel \boldsymbol{u} \parallel_2^2 + C \sum_{i=1}^{N} \xi_i \tag{3}$$

We develop the following optimization job, where C > 0 is penalty parameter for data points to create ideal hyperplane w $\times\cdot\phi$(x) + b = 0 by Equation (4).

$$\min_{\mathbf{w},b,\varepsilon_i} \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C \sum_{i=1}^{n} \varepsilon_i \tag{4}$$

Hence, the trade-off between ww^2 and $\sum_{i=1}^{n}\varepsilon_i$ is determined by the constant C and the slack variable ε_i. The aforementioned optimization issue is similar to the following under KKT conditions by Equation (5):

$$\min\frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) - \sum_{i=1}^{n} a_i \tag{5}$$

where $K(x_i, x_j) = \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j})$ is an inner product in feature space. We are able to get w and b by resolving the previous issue. The decision function is then expressed as Equation (6):

$$f(\mathbf{x}, \mathbf{w}, b) = \text{sgn}\ (\mathbf{w}\cdot\phi(\mathbf{x}) + b) = \text{sgn}\left(\sum_{i=1}^{n} a_i y_i K(\mathbf{x_i}, \mathbf{x}) + b\right) \tag{6}$$

The unseen sample x is assigned to Class 1 if f(x, w, b) is positive; else, x is assigned to Class 1. We can see that in the case of SVM, the dual issue and the decision function are just wholly linked to the kernel of samples. Both histograms have m bins, and the $j_b{}^{th}$ bin is represented by x_{1j_b} and x_{2j_b} for $j_b = 1, \ldots, m$. In the event when x_1 and x_2 are both N pixels in size, we have $\sum\sum_{j_b=1}^{m} x_{1j_b} = N$ and $\sum_{j_b=1}^{m} x_{2j_b} = N$. The following equation is used to determine the histogram intersection by Equation (7):

$$K_{\text{HIK}}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j_b=1}^{m} \min\left\{x_{1j_b}, x_{2j_b}\right\} \tag{7}$$

The Hellinger's kernel x_1 and x_2, the χ^2 kernel are calculated as Equations (8) and (9).

$$K_{\chi^2}(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j_b=1}^{m} \frac{\left(x_{1j_b} - x_{2j_b}\right)^2}{x_{1j_b} + x_{2j_b}} \tag{8}$$

$$K_H(\mathbf{x}_1, \mathbf{x}_1) = \sum_{j_b=1}^{m} \sqrt{x_{1j_b} x_{2j_b}} \tag{9}$$

The single-kernel SVM model FSK is written as follows, given a set of samples $\{x_i, y_i\}$ N_i=1 where xi is input vector and y_i is its class label by Equation (10):

$$f_{SK}(x) = \sum_{i=1}^{N} \alpha_i k(x_i, x) + b \tag{10}$$

where (α1, ..., αN) is weight vector, k(·) is kernel function, and b is bias. To implement SVM, many kernel functions are used. The global kernel as well as local kernel are two categories for these kernel functions with various characteristics. High-frequency time series demand a local kernel function with strong local learning capabilities. On the other hand, low-frequency time series demand a global kernel function with strong global learning capabilities. The properties of the data time series are taken into account when choosing the appropriate kernel function. The model's capacity for prediction can be increased by picking the right kernel function. Gaussian kernel k_{GAU}, polynomial kernel k_{POL}, and linear kernel k_{LIN} are some of several kernel functions by Equations (11) and (12):

$$k_{\text{UN}}\left(x_i, x_j\right) = \left\langle x_i, x_j \right\rangle \tag{11}$$

$$k_{POL}\left(x_i, x_j\right) = \left(\left\langle x_i, x_j \right\rangle + 1\right)^q$$

q is natural number.

$$k_{GAU}\left(x_i, x_j\right) = \exp\left(\frac{-\parallel x_i - x_j \parallel_2^2}{2s^2}\right), s > 0 \tag{12}$$

Different learning capacities exist among these three categories of kernel functions. The ACF can reflect the lag, which is typically present in time series forecasting. In contrast to high-frequency time series, low-frequency time series have a different lag. In general, as frequency rises, a time series' decomposed component's time lag shortens. More

complex time series with short lags call for methods with high local learning capabilities. On the other hand, time series with significant lags call for models with robust global learning capabilities.

S_b null space's low-dimensional complement space, designated B, is first obtained. Assuming that S_b and S_w are the scatter matrices between and within classes, respectively. Let V_b be the M eigenvectors of S_b that correspond to M non-zero eigenvalues = [b_1, ..., b_M] and $M = min(C\ 1, J)$. As a result, V_b extends across the S_b subspace B and is scaled by $U = V_{b1}/2\ b$ to produce $UT_{Sb}U = I$, where I denotes the identity matrix (M M) and b = diag() denotes the diagonalization operator. To obtain the relevant feature representations, all training samples z_{ij} are first projected into the subspace spanned by U, where yij is the feature representation of z_{ij} in the subspace B. This prepares all training samples z_{ij} for QDA in B by Equation (13).

$$\widehat{\mathbf{\Sigma}}_i(\alpha, \gamma) = \left[(1-\gamma)\widehat{\mathbf{L}}_i(\alpha) + \frac{7}{M}\mathrm{tr}\widehat{\mathbf{\Sigma}}_i(\alpha)\right]\mathbf{I} \tag{13}$$

where the prior probability for class i is $\pi_i = C_i/N$. The suggested approach entails minimising a multivariate quadratic function subject to linear constraints in the manner described Equations (14)–(16).

$$\min_x \frac{1}{2}x^T Qx - F^T x \tag{14}$$

$$\mathrm{st} x_i \geq 0 \forall i = 1 \ldots M \tag{15}$$

$$\| x \|_1 = 1 \tag{16}$$

where x is a d-dimensional vector, Q is a symmetric positive semidefinite matrix, and F is an entry-free vector in R d. Redundancy among variables is represented by Q, and F gauges how closely each feature is related to the target class (relevance). We decided to normalise each feature's contribution because the components of the solution vector x∗ represent the weight of each feature as shown in Equation (17).

$$\delta(x) = x^T\left(\sum\nolimits_1 - \sum\nolimits_2\right)^{-1}x + 2\left(\sum\nolimits_2^{-1}\ \mu_2 - \sum\nolimits_1^{-1}\ \mu_1\right)^T x \tag{17}$$

3.2. Adversarial Bayesian Belief Networks Based Malicious Attack Detection

To drive the error into the $\left(\mathcal{O}(\varepsilon) + \frac{2\mathrm{na}\sigma^2}{\mu}\right)$ neighbourhood of the optimum7, α or to achieve by Equation (18), let us identify the parameters p that lead to the fastest rate.

$$\mathbb{E}\left[\| x^k - x(\lambda) \|^2\right] \leq \varepsilon \| x^0 - x(\lambda) \|^2 + \frac{2\mathrm{ma}\sigma^2}{\mu} \tag{18}$$

The parameter $p^* = \frac{\lambda}{L+\lambda}$ reduces the predicted number of communications for attaining as well as the number of repetitions. For example, the ideal number of iterations is $2\frac{L+\lambda}{\mu}\log\frac{1}{\varepsilon}$, and the ideal number of communications to expect is $\frac{2\lambda}{\lambda+L}\frac{L}{\mu}\log\frac{1}{\varepsilon}$. We employ the relativistic average discriminator D_{Ra} to render the output image virtually identical to the original. Equation (19) represents the objective functions.

$$\mathcal{L}_{R_{a\_}D} = -\mathbb{E}_X[\log(D_{R_a}(x))] - \mathbb{E}_{x,\lambda,}[\log(1 - D_{Ra}(G(x, v, c)))] \tag{19}$$

The likelihood that the produced image is more real than the real image can be maximised by minimising the loss $\mathcal{L}_{R_{a\_}D}$. We subject the generator to a cycle consistency loss, denoted by Equation (20):

$$\mathcal{L}_{cyc} = \mathbb{E}_{x,p,\in}[\| x - G(G(x, v, c), v, 1-c) \|_1] \tag{20}$$

The source of the image is then determined by layering an auxiliary classifier called Dind on top of the discriminator network. The following paired adversarial loss by Equation (21) is included to further ensure fitting of picture translation method:

$$\mathcal{L}_{\text{pis}} = -\mathbb{E}_{x_{0,},x_v}\left[\log\left(D_{pis}(x_{0,}x_v)\right)\right] - \mathbb{E}_{x,0,f}\left[\log\left(1 - D_{\text{pis}}\ (x, G(x,v,c))\right)\right] \tag{21}$$

D_{pis} is employed in this situation to determine whether two photos belong to the same class. Our objective is to remove variation v from input image xv using operation (v, c = 10). To accomplish this, we layer an additional classifier called D_{var} on top of discriminator network to identify various types of variation in images. Classification loss during training discriminator network is as Equation (22):

$$\mathcal{L}^{\tau}_{\text{our}} = -\mathbb{E}_{x,p}[\log(D_{\text{var}}\ (v \mid x))] \tag{22}$$

Discriminator network may categorise real image x into variant type v by minimizing the aforementioned formula. The following Equation (23) is utilized to represent the final output image,

$$x_{\text{out}} = x + \left(x_f - x\right) \odot x_m \tag{23}$$

element-wise product is $\odot$ located where. The following equation is added for the mask x_m (24):

$$\mathcal{L}_{\text{mask}} = \left(\frac{1}{W}\sum_k |x_m|k\ \|\right)^2 \tag{24}$$

where W is number of pixels and $x_m\,|\,k\,|$ is k-th pixel of x_m. The formula shown above promotes minimising alterations to the source image. For computing unbiased estimates of, L, with one w.r.t. p under reparameterization ELBO is equivalent to (25):

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \tag{25}$$

where $\mathbf{z} = g(\epsilon, \boldsymbol{\phi}, \mathbf{x})$. As a result, using a single noise sample obtained ϵ from $p(\epsilon)$ by Equations (26)–(28), we may create a straightforward Monte Carlo estimate $\widetilde{\mathcal{L}}_{\theta,\phi}(\mathbf{x})$ of individual data point ELBO.

$$\epsilon \sim p(\boldsymbol{\epsilon}) \tag{26}$$

$$\mathbf{z} = \mathbf{g}(\phi, \mathbf{x}, \boldsymbol{\epsilon}) \tag{27}$$

$$\widetilde{\mathcal{L}}_{\theta,\phi}(\mathbf{x}) = \log p_\theta(\mathbf{x},\mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x}) \tag{28}$$

As a result, a structural learning strategy can be used to reduce the maximum in-degree. In practice, we examine the following equation's result: (29) Gi optimization to minimize specific class-to-feature arcs:

$$\mathcal{G}^*_i = \text{argmax}_{\mathcal{G}_i \subset \mathcal{G} \subset \mathcal{G}_i} \log P(\mathcal{G} \mid \mathcal{D}) \tag{29}$$

where it should be intended for sets of graphs to include one another in the arcs space, and by Equation (30):

$$\log P(\mathcal{G} \mid \mathcal{D}) = \sum_{i=1}^{n}\ \psi_a[C_i, \text{Pa}(C_i)] + \sum_{j=1}^{m}\ \psi_\alpha\left[F_j, \text{Pa}\left(F_j\right)\right] \tag{30}$$

according to $\mathcal{G}$, $\text{Pa}\left(F_j\right)$ denotes F_j parents, whereas $\text{Pa}(C_i)$ denotes C_i parents. Additionally, a BDEu score with the same sample size is available, where the first sum includes all of its parent states and the second sum includes all of F_j possible states. Additionally, the number of records required to ensure that F_j is in its kth state and that its parents are in their ith configuration is N_{ji}, which is equal to $\text{P}_\text{k}\text{N}_{\text{jik}}$. This indicates that the first sum on the right side remains constant. Therefore, the optimization in (31) can be achieved by

only considering the features. A feature's parents set can be chosen from any subset of C, reducing the problem to m separate local optimizations. G asserts that F_j parents are in fact.

$$\mathbf{C}_{F_j} = \arg_{\mathrm{Pa}(F_j)\subseteq \mathbf{C}} \psi_a\left[F_j, \mathrm{Pa}\left(F_j\right)\right] \tag{31}$$

Bipartite separation of class events and features makes this possible for each time j = 1, but directed cycles are typically found in a graph that maximizes all local scores. Assume that k is number of mixture components, that X is set of query variables, that Z is other variables, and that is the number. By equating C and Z, we can determine marginal distribution of X (32):

$$P(X = x) = \sum_{c=1}^{k} \sum_{z} P(C = c, X = x, Z = z) \tag{32}$$

where previous equality holds true because, for any j, $j\ 1\ zj\ P\ c\ z$. As a result, it is straightforward to disregard non-query variables Z when calculating $P(X = x)$, and regardless of $|Z|$, the calculation of $P(X = x)$ takes $O(|X|k)$. In contrast, Bayesian network inference is worst-case exponential in $|Z|$. By Equation (33), the visible unit x and the hidden layers of length l make up the joint distribution.

$$p\left(x, h^1, \cdots, h^{\updownarrow}\right) = p\left(h^{\updownarrow-1}, h^{\updownarrow}\right)\left(\prod_{k=1}^{\updownarrow-2} p\left(h^k \mid h^{k+1}\right)\right) p\left(x \mid h^1\right) \tag{33}$$

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[-\rho \log p\left(x^{(i)}; \theta_{DBN}\right)\right. \tag{34}$$

Remember that layer-wise updating necessitates fixing every problem from the bottom hidden layer to the top visible layer. The following optimization issue is fixed by Equation (35) for the fine-tuning phase.

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}\left(\theta_L; y^{(i)}, h\left(x^{(i)}\right)\right) - \rho \log p\left(x^{(i)}; \theta_{DBN}\right)\right] \tag{35}$$

where the classifier's parameters are, L() is a loss function, and h denotes the final hidden features at layer l. For the sake of simplicity, we will set $h\left(x^{(i)}\right) = h\left(x_{\updownarrow}^{(i)}\right)$. We first aggregate training and fine-tuning goals using a simple model. The model's definition(DBN+loss) is given by Equation (36),

$$\min_{\theta_L, \theta_{DBN}} E_{y,x}[\mathcal{L}(\theta_L; \mathbf{y}, h(\mathbf{x}))] + \rho E_{\mathbf{x}}[-\log p(\mathbf{x}; \theta_{DBN})] \tag{36}$$

based on training samples D, and experimentally by Equation (37),

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}\left(\theta_L; y^{(i)}, h\left(x^{(i)}\right)\right) - \rho \log p\left(x^{(i)}; \theta_{DBN}\right)\right] \tag{37}$$

where the underlying parameters are θ_L, θ_{DBN}. We initially develop an anticipated loss model using the conditional distribution $p(h|x)$ generated by DBN. This paradigm is used to classify the hidden space. Because it reduces the expected loss, it should be more dependable and, as a result, produce better accuracy on data that has not been observed. The attack detection model is given by Figure 2.

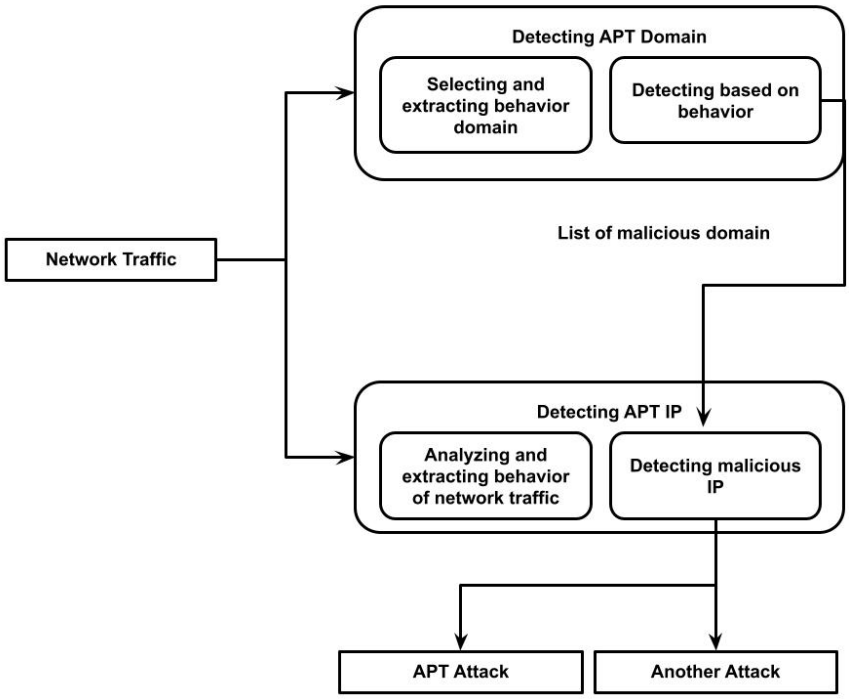

Figure 2. Architecture of proposed attack system.

4. Experimental Analysis

On a server running a 32-bit operating system at 2.80 GHz, a Core E7400 processor, 3.00 GB of RAM, and the proposed architecture with fog and cloud nodes are tested.

Dataset description: Although many of those datasets are still kept private, mostly for security reasons, some of them, such DARPA 98, KDD99, and UNSW-NB15, are now open to the public. Although many datasets have been created, there have not been many realistic IoT and network traffic datasets that incorporate fresh Botnet instances. What is more, some databases do not include IoT-generated traffic, and others do not add any new features.

A set of examples used to adjust a classifier's hyperparameters, or architecture, is called a validation dataset. Development set or "dev set" are other names for it. For artificial neural networks, number of hidden units in each layer is an example of a hyperparameter. The hyperparameter tuning process makes use of the validation set. The best model is ultimately evaluated using the test set. If hyperparameter tuning is not going to be carried out, then the validation set is redundant and not required.

Table 2 analysis is based on various malicious attack datasets. Here, the datasets analysed are DARPA 98, KDD99,UNSW-NB15 dataset. The parametric analysis is carried out in terms of throughput, data traffic analysis, end-end delay, packet delivery ratio, energy efficiency, and QoS.

Table 2. Analysis based on various malicious attack datasets.

Dataset	Techniques	Throughput	Data Traffic Analysis	End-End Delay	PDR	Energy Efficiency	QoS
DARPA 98	**SVM**	89	59	45	81	82	71
	SYN-DOS	92	62	44	83	85	75
	CS_NTA_MADML	93	63	42	85	88	77
KDD99	**SVM**	92	65	48	82	89	72
	SYN-DOS	94	68	46	84	92	74
	CS_NTA_MADML	96	72	44	86	93	76
UNSW-NB15	**SVM**	95	58	52	85	85	75
	SYN-DOS	96	72	50	88	88	77
	CS_NTA_MADML	98	74	45	92	92	79

The Figure 3a–f shows the analysis for DARPA 98 dataset. The proposed technique attained throughput of 93%, data traffic analysis of 63%, end-end delay of 42%, packet delivery ratio of 85%, energy efficiency of 88%, QoS of 77%; existing SVM attained throughput of 89%, data traffic analysis of 59%, end-end delay of 45%, packet delivery ratio of 81%, energy efficiency of 82%, QoS of 71%; and SYN-DOS attained throughput of 92%, data traffic analysis of 62%, end-end delay of 44%, packet delivery ratio of 83%, energy efficiency of 85%, QoS of 75%.

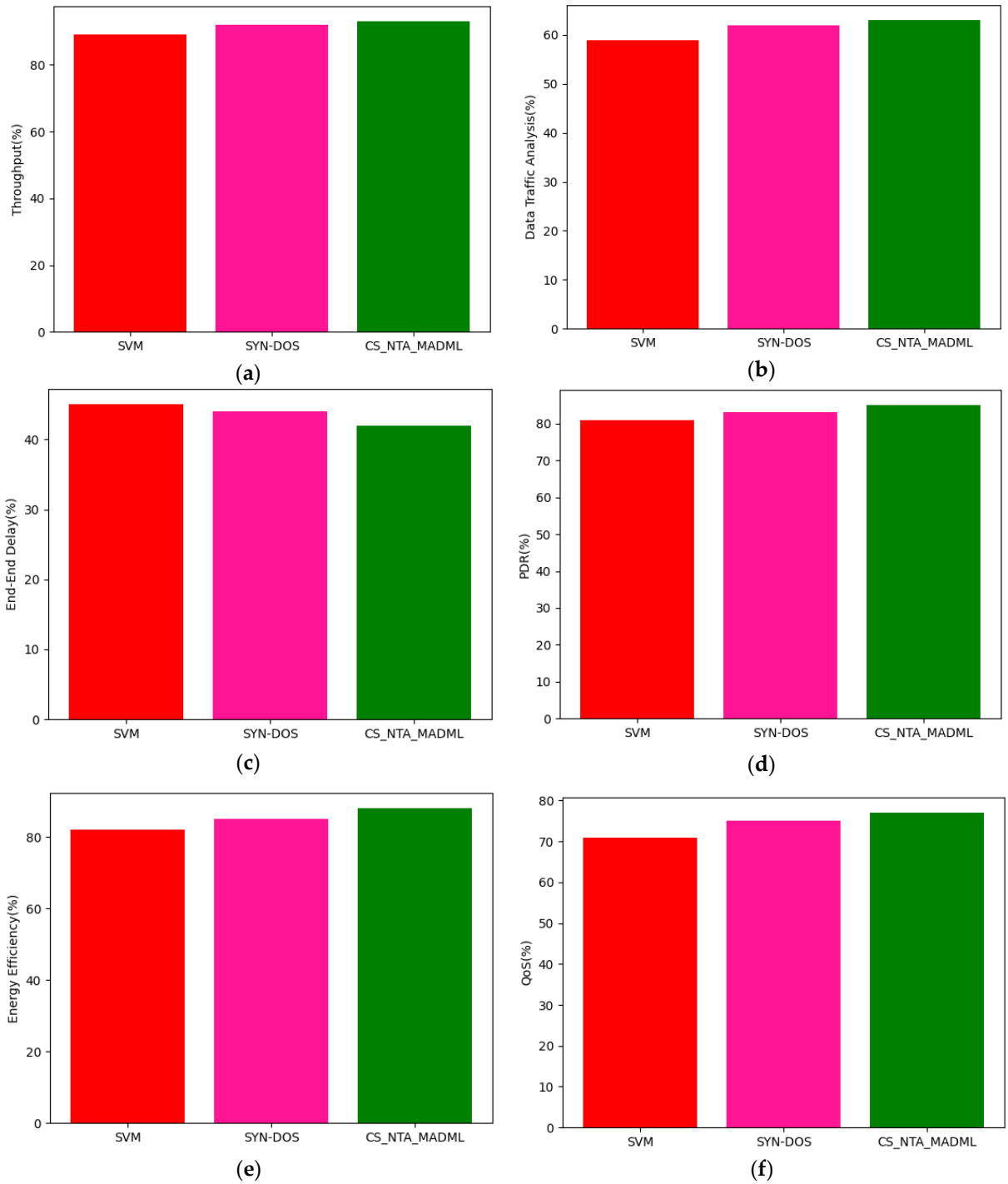

Figure 3. Analysis for DARPA 98 dataset in terms of (**a**) throughput, (**b**) data traffic analysis, (**c**) end-end delay, (**d**) packet delivery ratio (PDR), (**e**) energy efficiency, and(**f**) QoS.

The Figure 4a–f shows a KDD99 dataset based comparative analysis between the proposed and existing techniques. The proposed technique attained throughput of 96%, data traffic analysis of 72%, end-end delay of 44%, packet delivery ratio of 86%, energy efficiency of 93%, QoS of 76%; existing SVM attained throughput of 92%, data traffic analysis of 65%, end-end delay of 48%, packet delivery ratio of 82%, energy efficiency of 89%, QoS of 72%; and SYN-DOS attained throughput of 94%, data traffic analysis of 68%, end-end delay of 46%, packet delivery ratio of 84%, energy efficiency of 92%, QoS of 74%.

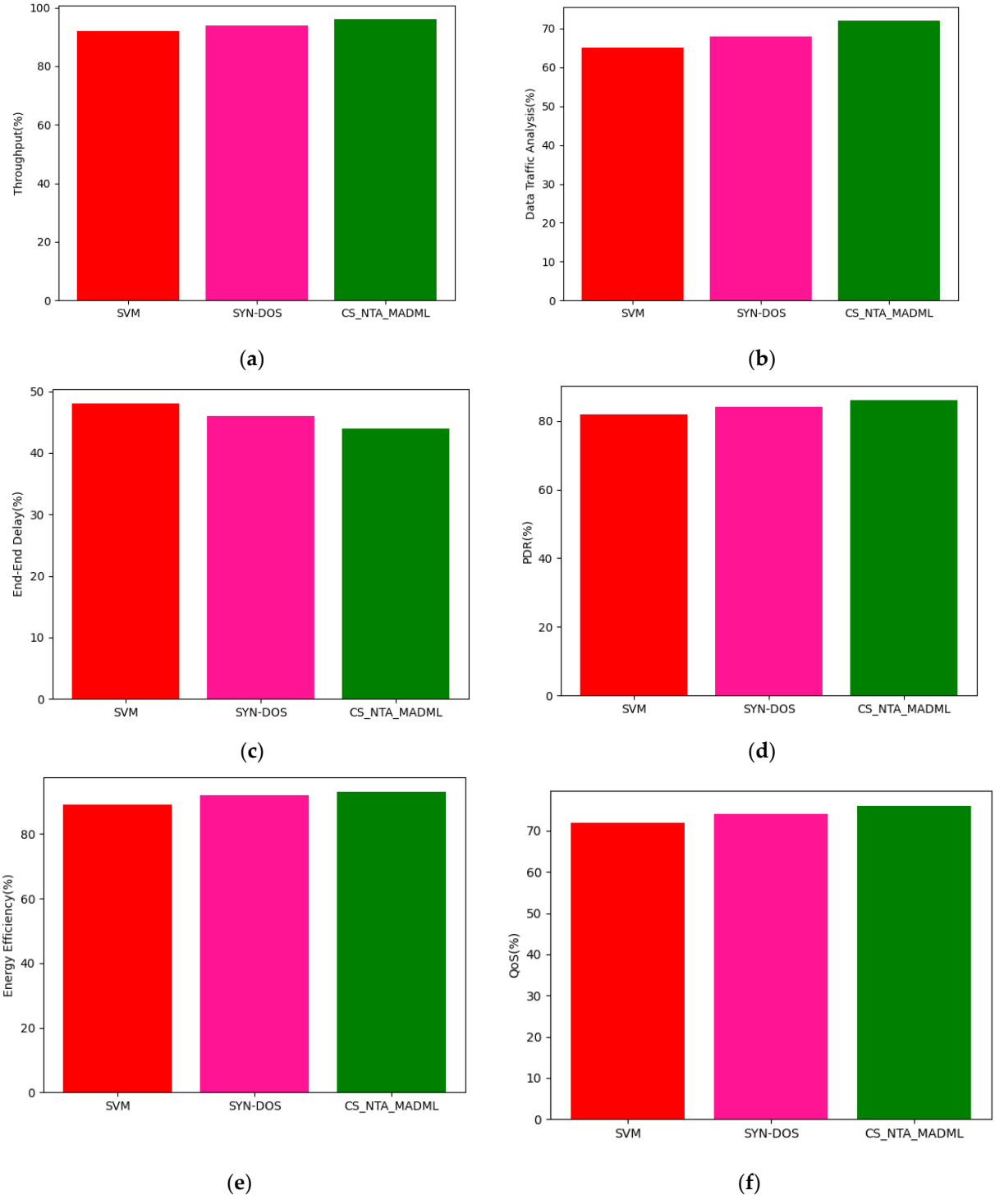

Figure 4. Analysis for KDD99dataset in terms of (**a**) throughput, (**b**) data traffic analysis, (**c**) end-end delay, (**d**) packet delivery ratio (PDR), (**e**) energy efficiency, and(**f**) QoS.

The Figure 5a–f analysis for UNSW-NB15 dataset. The proposed technique attained throughput of 98%, data traffic analysis of 74%, end-end delay of 45%, packet delivery ratio of 92%, energy efficiency of 92%, and QoS of 79%; existing SVM attained throughput of 95%, data traffic analysis of 58%, end-end delay of 52%, packet delivery ratio of 85%, energy efficiency of 85%, and QoS of 75%; and SYN-DOS attained throughput of 96%, data traffic analysis of 72%, end-end delay of 50%, packet delivery ratio of 88%, energy efficiency of 88%, QoS of 79%.

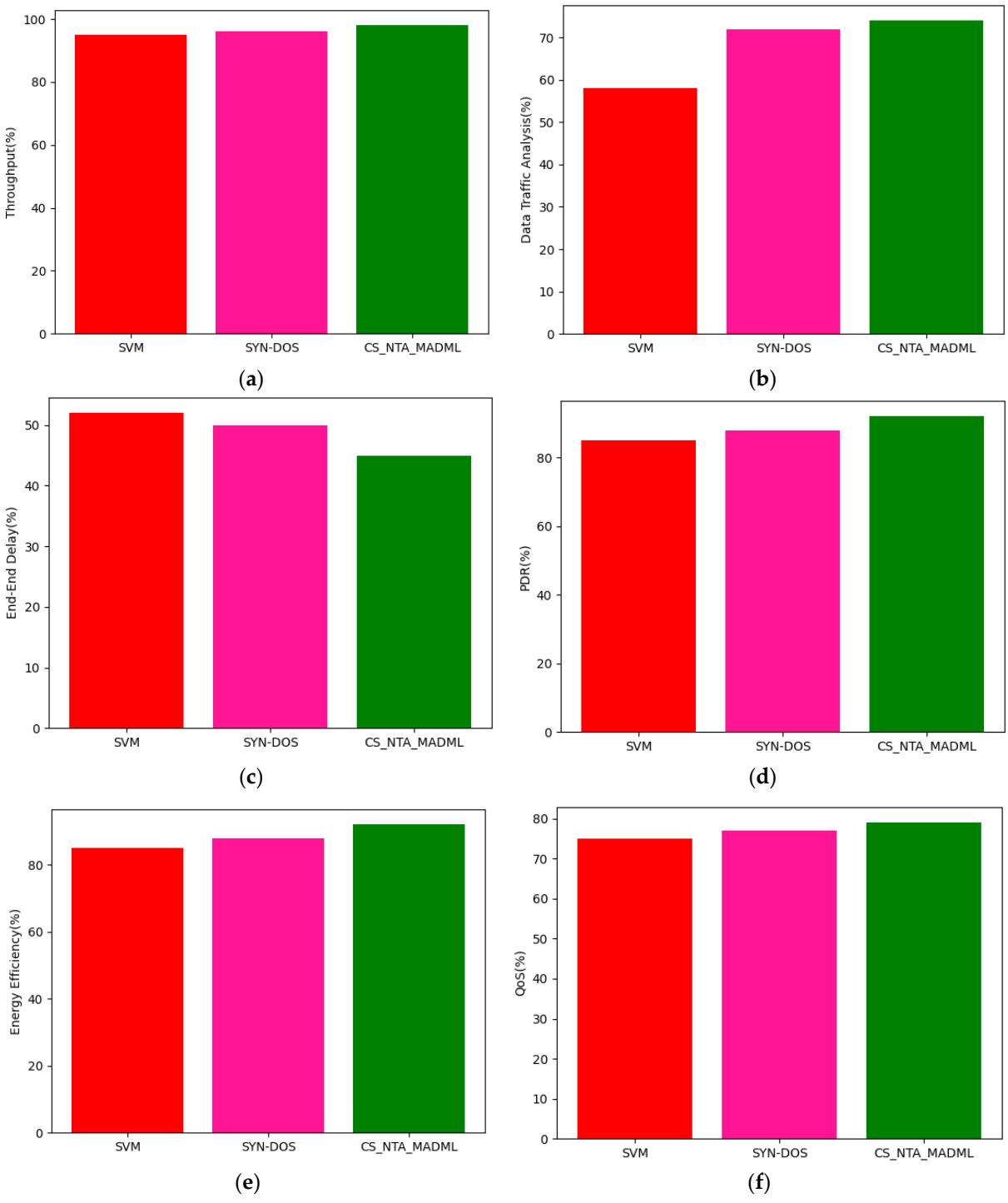

Figure 5. Analysis for UNSW-NB15 dataset in terms of (**a**) throughput, (**b**) data traffic analysis, (**c**) end-end delay, (**d**) packet delivery ratio(PDR), (**e**) energy efficiency, and(**f**) QoS.

5. Discussion

Different combinations of features are obtained when cyber virus attacks are detected utilising network traffic features and neural networks. For the purpose of learning, this study employs a dataset containing 442,240 data points that combines existing datasets with the findings of laboratory trials. It is advised that malware in IoT devices be detected using the current neural network model. With a lower false alarm rate, the system can identify aberrant network activity and create alarms for it. We evaluated the binary categorization of network traffic using the DARPA 98, KDD 99, and UNSW-NB15 datasets. The outcomes demonstrated that association rule-based filtering might significantly increase the system's detection precision. In addition, our detection method performed well in an experimental setting with multiple classes. In terms of detection results, this two-level detection system that first classifies and then filters network traffic provides higher precision and fewer false positives.

6. Conclusions

This research proposes a novel method in cybersecurity based on IoT artificial intelligence techniques for a sustainable smart city. A traffic analysis has been carried out using a kernel quadratic vector discriminant machine which enhances the data transmission by reducing network traffic and the malicious attack detection is carried out using adversarial Bayesian belief networks. The proposed technique attained throughput of 98%, data traffic analysis of 74%, end-end delay of 45%, packet delivery ratio of 92%, energy efficiency of 92%, and QoS of 79%.A deep neural network's structure still has a lot of space for improvement, and future work can solve the difficulty of boosting precision while maintaining recall. The proposed method will be expanded in the future to incorporate information from other attack kinds and sources to improve its capacity for making decisions and to counter future attempts. Studying a network evolutionary algorithm, such as the imperialist competitive algorithm, is thought to be of utmost importance for future research on complementing the proposed technique.

Author Contributions: All authors have equal contributions. All authors have read and agreed to the published version of the manuscript.

Funding: The Research Groups Funding program grant (NU/RG/SERC/12/26), the Deanship of Scientific Research, Najran University, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be provided upon request.

Acknowledgments: The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program grant code (NU/RG/SERC/12/26).

Conflicts of Interest: The authors declare that they have no conflict of interest.

Notations

List of Notations Used	Meaning
$\mathbb{R}^n$	Feature space
γ	Class label set
$\mathbb{C}$	Base classifier
H	Proposed classifier
η	Number of training examples
$\mathrm{DT} = \{P_i, Y_i\}_{i=1}^{n} (P_i \in \mathbb{R}^n, Y_i \in \gamma)$	Training dataset
K	Divide $\mathbb{DT}$ into K equal parts subset

References

1. Gao, Z.; Fang, S.C.; Luo, J.; Medhin, N. A kernel-free double well potential support vector machine with applications. *Eur. J. Oper. Res.* **2021**, *290*, 248–262. [CrossRef]
2. Xie, Z.; Xu, Y.; Hu, Q. Uncertain data classification with additive kernel support vector machine. *Data Knowl. Eng.* **2018**, *117*, 87–97. [CrossRef]
3. Inayat, U.; Zia, M.F.; Mahmood, S.; Khalid, H.M.; Benbouzid, M. Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods 2022, Analysis, and Future Prospects. *Electronics* **2022**, *11*, 1502. [CrossRef]
4. Do Xuan, C.; Dao, M.H. A novel approach for APT attack detection based on combined deep learning model. *Neural Comput. Appl.* **2021**, *33*, 13251–13264. [CrossRef]
5. Inayat, U.; Ali, F.; Khan, H.M.A.; Ali, S.M.; Ilyas, K.; Habib, H. Wireless Sensor Networks: Security, Threats, and Solutions. In Proceedings of the 2021 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 9–10 November 2021; pp. 1–6.
6. Inayat, U.; Zia, M.F.; Ali, F.; Ali, S.M.; Khan, H.M.A.; Noor, W. Comprehensive review of malware detection techniques. In Proceedings of the 2021 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 9–10 November 2021; pp. 1–6.
7. Zagrouba, R.; Alhajri, R. Machine Learning based Attacks Detection and Countermeasures in IoT. *Int. J. Commun. Netw. Inf. Secur.* **2021**, *13*, 158–167. [CrossRef]
8. Salih, A.; Zeebaree, S.T.; Ameen, S.; Alkhyyat, A.; Shukur, H.M. A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection. In Proceedings of the 2021 7th International Engineering Conference "Research & Innovation amid Global Pandemic"(IEC), Erbil, Iraq, 24–25 February 2021; pp. 61–66.
9. Do Xuan, C. Detecting APT attacks based on network traffic using machine learning. *J. Web Eng.* **2021**, *20*, 171–190. [CrossRef]
10. Xuan, C.D.; Duong, D.; Dau, H.X. A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic. *J. Intell. Fuzzy Syst.* **2021**, *40*, 11311–11329. [CrossRef]
11. Anusha, M.; Karthika, M. Investigation on Malware Detection Using Deep Learning Methods for Sustainable Development. In *Micro-Electronics and Telecommunication Engineering, Proceedings of theInternational Conference on Micro-Electronics and Telecommunication Engineering, Ghaziabad, India, 25–25 September 2021*; Springer: Singapore, 2021; pp. 581–592.
12. Novaes, M.P.; Carvalho, L.F.; Lloret, J.; Proença, M.L., Jr. Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. *Future Gener. Comput. Syst.* **2021**, *125*, 156–167. [CrossRef]
13. Shahid, W.B.; Abbas, H.; Aslam, B.; Afzal, H.; Khalid, S.B. A framework to optimize deep learning based web attack detection using attacker categorization. In Proceedings of the 2021 IEEE 19th International Conference on Embedded and Ubiquitous Computing (EUC), Shenyang, China, 20–22 October 2021; pp. 95–101.
14. Shahid, W.B.; Aslam, B.; Abbas, H.; Khalid, S.B.; Afzal, H. An enhanced deep learning based framework for web attacks detection, mitigation and attacker profiling. *J. Netw. Comput. Appl.* **2022**, *198*, 103270. [CrossRef]
15. Strecker, S.; Dave, R.; Siddiqui, N.; Seliya, N. A modern analysis of aging machine learning based IOT cybersecurity methods. *arXiv* **2021**, arXiv:2110.0783. [CrossRef]
16. AlZubi, A.A.; Al-Maitah, M.; Alarifi, A. Cyber-attack detection in healthcare using cyber-physical system and machine learning techniques. *Soft Comput.* **2021**, *25*, 12319–12332. [CrossRef]
17. Waqas, M.; Kumar, K.; Laghari, A.A.; Saeed, U.; Rind, M.M.; Shaikh, A.A.; Hussain, F.; Qazi, A.Q. Botnet attack detection in Internet of Things devices over cloud environment via machine learning. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6662. [CrossRef]
18. Khan, M.A. HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes* **2021**, *9*, 834. [CrossRef]
19. Sarker, I.H. Deep cybersecurity: A comprehensive overview from neural network and deep learning perspective. *SN Comput. Sci.* **2021**, *2*, 154. [CrossRef]
20. Karthika, R.A.; Maheswari, M. Detection analysis of malicious cyber attacks using machine learning algorithms. *Mater. Today Proc.* **2022**, *68*, 26–34. [CrossRef]
21. Sahu, A.K.; Sharma, S.; Tanveer, M.; Raja, R. Internet of Things attack detection using hybrid Deep Learning Model. *Comput. Commun.* **2021**, *176*, 146–154. [CrossRef]
22. Ullah, S.; Khan, M.A.; Ahmad, J.; Jamal, S.S.; e Huma, Z.; Hassan, M.T.; Pitropakis, N.; Arshad; Buchanan, W.J. HDL-IDS: A hybrid deep learning architecture for intrusion detection in the Internet of Vehicles. *Sensors* **2022**, *22*, 1340. [CrossRef] [PubMed]
23. Ravi, V.; Pham, T.D.; Alazab, M. Attention-based multidimensional deep learning approach for cross-architecture IoMT malware detection and classification in healthcare cyber-physical systems. *IEEE Trans. Comput. Soc. Syst.* **2022**. [CrossRef]
24. Al-Haija, Q.A. Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. *Front. Big Data* **2021**, *4*, 782902. [CrossRef] [PubMed]
25. Mihoub, A.; Fredj, O.B.; Cheikhrouhou, O.; Derhab, A.; Krichen, M. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Comput. Electr. Eng.* **2022**, *98*, 107716. [CrossRef]