Intelligent system for IoT botnet detection using SVM and PSO optimization

Mahmoud A. Salam

Faculty of Computers and Information, Mansoura University, Egypt

Email: {maasalam99@yahoo.com}

Abstract: Botnet attacks involving Internet-of-Things (IoT) devices have skyrocketed in recent years due to the proliferation of internet IoT devices that can be readily infiltrated. The botnet is a common threat, exploiting the absence of basic IoT security technologies and can perform several DDoS attacks. Existing IoT botnet detection methods still have issues, such as relying on labeled data, not being validated with newer botnets, and using very complex machine learning algorithms, making the development of new methods to detect compromised IoT devices urgent to reduce the negative implications of these IoT botnets. Due to the vast amount of normal data accessible, anomaly detection algorithms seem to promise for identifying botnet attacks on the Internet of Things (IoT). For anomaly detection, the One-Class Support vector machine is a strong method (ONE-SVM). Many aspects influence the classification outcomes of the ONE-SVM technique, like that of the subset of features utilized for training the ONE-SVM model, hyperparameters of the kernel. An evolutionary IoT botnet detection algorithm is described in this paper. Particle Swarm Optimization technique (PSO) is used to tune the hyperparameters of the ONE-SVM to detect IoT botnet assaults launched from hacked IoT devices. A new version of a real benchmark dataset is used to evaluate the proposed method's performance using traditional anomaly detection evaluation measures. This technique exceeds all existing algorithms in terms of false positive, true positive and rates, and G-mean for all IoT device categories, according to testing results. It also achieves the shortest detection time despite lowering the number of picked features by a significant amount.

Keywords: Botnets, IOT, Particle Swarm, Anomaly detection, Feature selection, One class support vector machine, Intrusion detection system.

1. Introduction

The Internet of Things (IoT) is a network of intelligent gadgets. Several applications, like traffic monitoring services and healthcare, use it in our daily lives [1]. Connecting a large number of devices to the Internet and allowing them to actively engage in the network, IoT allows machine-to-machine communication over the Internet [2]. In 2017, there were roughly 27 billion IoT devices, and by 2020, this number is anticipated to reach 50 billion [3]. There are a lot of security issues with IoT settings since they are generally made up of a lot of heterogeneous and low-cost devices that have little or no security built-in [4]. Growth in IoT devices and the lack of security have attracted unscrupulous individuals to launch a series of DDoS assaults through the deployment of massive IoT botnets [1]. One or more malicious people operate a botnet, which is a group of hacked devices used to carry out harmful actions. Among the most

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

prevalent IoT botnet families are BASHLITE and Mirai [5]. IoT botnets are characterized by the following major features:

- Infected IoT devices are scattered all over the world, and the majority of them are Linux-based and live in the RAM of compromised IoT devices.
- Because IoT botnets are usually not harmful to the hacked IoT devices, it is difficult to identify these botnets.

Our lives have already been negatively affected by IoT botnets, with the Mirai attack in 2016 serving as the most notable example yet. After infecting surveillance cameras—an IoT device—by exploiting the default security settings on them, a botnet dubbed Mirai launched a massive DDoS attack against Dyn, a major domain service provider [6]. Furthermore, infected IoT devices may still not show any obvious indicators of infection, allowing them to carry on with their normal operations without being detected as compromised. Finding hacked IoT devices is a difficult task that requires particular techniques [5]. To protect the Internet of Things (IoT), the research community has identified the need for an Intrusion Detection System (IDS) [7]. Abuse detection and anomaly detection are two types of IDS that can be used in conjunction. To discover anomalies, it's assumed that normal behaviour differs from the anomalous activity and that this difference may be observed. The system is good at detecting unknown threats, but it has a high rate of false alarms [8]. While on the other side, abuse detection can only identify well-known attacks by matching their behaviours of recorded network data. A high detection rate for known assaults and also a very low performance in detecting novel attacks are the results of this software's performance. The vast majority of IDS for IoT nodes, however, suffer from several flaws, such as depending on tagged data, not being verified with updated botnets, and relying on quite complicated machine learning methods. The ONE-SVM technique [9] and PSO [10] are used to develop an evolutionary network-based algorithm for detecting IoT botnets in this research. ONE- SVM is a strong machine learning method that may be used to create a model using existing data to discover its usual behaviour. Any divergence from this usual behaviour is classified as anomalous by ONE-SVM. A botnet is a collection of IoT devices that have been compromised, and as a result, they behave uniquely. When it comes to detecting IoT botnets, this strategy is particularly useful because it reduces the cost of labelling and the need for harmful data in the training phase. The behaviour of IoT bots can also be detected by modelling regular traffic for each IoT device type since each IoT device has unique capabilities. However, ONE-SVM is highly sensitive to the starting values of its hyperparameters [11]. In the radial basis function (RBF), these hyperparameters include v and the kernel parameter λ. In addition, the attributes that are used to construct the ONE-SVM model are also crucial. Because of this, the algorithm's goal is to minimise the number of features selected by the algorithm to build an accurate general model and save running time [12]. The metaheuristic algorithm PSO, which was inspired by the way swarms hunt in nature. Due to its unique population structure, PSO has demonstrated exceptional performance potential. Since PSO is so efficient, it may be used in a variety of applications such as machine learning and engineering as well as wireless sensor networks and image processing [13]. To detect IoT botnet assaults, the proposed technique employs PSO for the first time in conjunction with ONE-SVM. While optimizing the ONE-SVM hyperparameters, PSO is also used for feature selection. In the proposed method, the key goals are to maximize TPR, minimize FPR, and reduce the number of selected features to the absolute minimum. On a new edition of the Network-based Detection of IoT (N-BaIoT) dataset, the PSO-ONE SVM will be assessed. OCSVM, Isolation forest (IF), GWO-OCSVM, ONE-SVM will be compared with the suggested algorithm to prove its efficacy. The following are some of the benefits of utilising the suggested method to detect compromised IoT systems:

- Heterogeneity: The IoT ecosystem is vastly different from traditional computer environments [7] [14]. Due to the variety of IoT devices, the suggested approach models each device separately.
- There is no supervision in the proposed algorithm. As a result, it has only been trained on unlabeled normals. Determining if a deviation from the regular behaviour is malicious is a result

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

of the model's ability to capture typical behaviour. This means that both known and undiscovered botnet attacks can be detected by the algorithm provided here. Because IoT botnets are constantly evolving[7], most anomaly detection techniques are ineffective [15].

- Since the suggested model is located on the gateway, it maintains the energy, compute, and memory resources on the restricted IoT devices. In this way, the functioning and lifespan of these IoT devices with limited capabilities will be preserved.

A new model based on PSO and ONE-SVM algorithms is proposed as the main contribution of this paper. Models such as the PSO and its unique hierarchical structure have been incorporated into the suggested model. That's the first time we've tried to use PSO to optimize ONE-SVM in order to detect IoT botnet attacks. This technique is also evaluated on real IoT botnets datasets, although most studies on IoT botnet assaults use simulated data.

Here's how the paper is structured, A review of prior work in the domain of anomaly detection approaches for IoT botnet assaults based on ONE_SVM is provided in Section 2. Section 3 summarises the current understanding of ONE-SVM and PSO algorithms. The suggested approach for detecting IoT botnet assaults is described in depth, along with the assessment measures that will be used to evaluate the algorithm. Sect. 4 describes the experiments and their outcomes. Sect. 5 summarises the results and discussion of the results of this research.

2. Related works

We shouldn't assume that all IoT device manufacturers will install the necessary security apps on their devices. IoT devices such as wearables have restricted access. Host-based anomaly detectors in IoT devices with limited computing, energy and memory resources will be used. Since IoT devices have a limited capability, these algorithms need to be lightweight to preserve the IoT devices' functioning. For conventional networks and IoT networks, [15] present a hierarchical classification of botnet detection algorithms.

According to [16], a hybrid of an artificial fish swarm algorithm with SVM can detect botnets. These studies sought to discover the most important botnet properties by evaluating data with optimized algorithms.

It was found that using darknet Telnet scans as information, [17] created a honeypot that emulated specific Telnet services. Using this honeypot, they were able to conduct a thorough analysis of ongoing attacks. There is no consensus on how to utilise honeypots to identify attacks on internet service providers in this study. Recently published research on network-based IoT botnet detection sparked this paper's development [18] [19]. And they created an actual IoT botnet dataset that was utilised as the basis for empirical testing. Instead of using a deep autoencoder, this work used far simpler algorithms widely used for anomaly detection to get greater G-mean, FPR and TPR over a fresh version of the N-BaIoT dataset and maybe improved performance.

As a result of Wolpert and Macready's No Free Lunch Theorem (NLF) [20], no optimization algorithm is effective than others for all optimization problems. Certain sorts of problems can be solved more efficiently with new algorithms. Swarm intelligence algorithms such as PSO have the potential to surpass traditional optimization techniques. As a result of its unique population structure not being present in other metaheuristics, this algorithm excels above others [13] [21]. To overcome the IoT botnet detection problem, it would be advisable to take advantage of PSO's special operators.

The botnet detection approach developed by [22] is based on unsupervised evolutionary Internet of Things (IoT). To identify between IoT botnet assaults that are initiated by compromised IoT devices, their proposed strategy had a primary objective. To do so, they used the Grey Wolf

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

Optimization method, which is an efficient swarm intelligence system (GWO). Their baseline One-Class Support Vector Machine was optimized using GWO (OCSVM). As a result of this, their model tended to uncover attributes that best explain the IoT botnet problem.

In conclusion, the suggested approach differs from earlier investigations because the proposed algorithm learns from normal data by constructing a model for each IoT device type, and then uses these models to identify both known and undiscovered IoT botnet assaults instantly.

3. Preliminaries

So that the suggested method can be better understood, this part briefly introduces the ONE-SVM in Sect. 3.1 and the PSO algorithm in Sect. 3.2.

3.1 ONE-SVM algorithm

Based on two-class SVM, Schölkopf suggested ONE-SVM [9]. A single class classification problem can be solved by using ONE-SVM, which separates target instances from outliers. Assume that *xL* for L = 1,...,n are the training examples in X, and that φ(xL) is the nonlinear transformation function that translates an instance *xL* from X to F. According to the concept of One-SVM, the training instances in F are separated with maximum margin from the origin in order to generate a *F* hyperplane corresponding to the kernel. The hyperplane is assumed to be defined by $\prod^{\star} : \langle w^{\star}, \varphi(x)\rangle - \varrho^{\star} = 0$ i, where w is the hyperplane's normal vector, and is the threshold. As demonstrated in Eq. (1), one can compute both w and p by solving the following optimization problem [9]. The hyperplane with the greatest margin can then be found.

$$\min_{w\in F,\xi\in\mathbb{R}^n,\rho\in\mathbb{R}} \frac{1}{2} \| \mathbf{w} \|^2 + \frac{1}{vn}\sum_i \xi_i - \rho \qquad (1)$$

where $\langle w, \boldsymbol{\rho}(xi)\rangle \geq \boldsymbol{\rho} - \boldsymbol{\zeta}i$, $\boldsymbol{\zeta}i \geq 0$, where $\boldsymbol{\zeta}$i are defined as the slack variables that embedded to an inequality constraint to map it to an equality that may embedded in some cases. If you're going to use ONE-SVM, you're going to need the option v = (0,1). Outlier percent is a measure of both the upper and lower bounds on the number of outliers among all instances. The ONE-SVM dual problem can be derived from the Lagrangian dual problem as demonstrated in Eq. (2):

$$\begin{aligned} &\max_{\alpha} -\frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{such that} \\ &0 \leq \alpha_i \leq \frac{1}{vn} \\ &\sum_{i=1}^{n} \alpha_i = 1 \end{aligned} \qquad (2)$$

Such that the kernel function is defined by k(xi , xj) with a value is $\langle \boldsymbol{\varphi}(xi , xj)\rangle$. Then after obtaining the optimal solution $\boldsymbol{\sigma}$, the threshold parameter $\boldsymbol{\rho}$ can be defined by adapting $\boldsymbol{\rho}$ with $\langle w, \boldsymbol{\varphi}(xi)\rangle$, where w equals to $\sum_{i=1}^{n} \boldsymbol{\alpha_i}\, \boldsymbol{\varphi}(\boldsymbol{x_i})$ and xi is somewhere instances with $\boldsymbol{\alpha}$i $\in (0, \frac{1}{vn})$. Not only that but also, the points with $\boldsymbol{\alpha_i} > 0$ are defined as support vectors. The ONE-SVM function f(x) is determined as shown in Eq. (3):

$$f(x) = \langle \mathbf{w}, \varphi(\mathbf{x}_i)\rangle - \rho = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \rho \qquad (3)$$

This instance is classified as positive with decision function is greater than or equal to zero; otherwise, it is classified as negative.

3.2 The particle swarm optimization (PSO) algorithm

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

Based on the social behaviour model, [23] created particle swarm optimization (PSO). As a result of space constraints, we will only briefly introduce PSO in this article. It varies from typical optimization approaches in that a population of possible solutions is used in the search process instead of just one. The search is guided by the direct fitness information, rather than function derivatives or related knowledge. PSO has the potential to solve the 2-D maximum entropy problem.

A set of random particles (solutions) is used to initialise PSO, which then seeks optimal solutions by updating successive generations of the algorithm. A particle's goal function values can be improved by using what other particles have discovered or experienced in their exploration and hunt for higher values. In the swarm, let me be the index of particles. They move at speeds of ***vi***, which are constantly changed according to their individual prior best solution ***si*** as well as the previous best solution s of the entire swarm, through an n-dimensional search space $\boldsymbol{R^n}$ A linear combination of location and velocity vectors is used to determine the updates to the velocity vectors. According to the following equations, the particles interact and move in different directions:

$$v_i^{(t+1)} = wv_i^{(t)} + c_1 r_1^{(t)} \left(s_i^{(t)} - p_i^{(t)}\right) + c_2 r_2^{(t)} \left(\hat{s}^{(t)} - p_i^{(t)}\right)$$
$$p_i^{(t+1)} = v_i^{(t+1)} + p_i^{(t)} \quad (4)$$

This corresponds to random values between zero and one (r^t1 and r^t2). There are two learning components c_1 and c_2. And w is the inertia weight of the system. By providing upper and lower constraints on ***vi***, it is possible to prevent particles from moving too fast in the search space. To discover the optimum, we can utilize a conventional approach to find it. When the maximum number of iterations is reached or the minimum error condition is met, the search operation stops.

According to industry standards, the standard procedure is as follows.

1. To do this, you must set the iteration number t to 0. Initiate randomly the swarm ***S*** of ***m*** particles (population number) so that each particle's position p0 meets the given parameters.
2. ***F(pi)***, the object function, should be used to determine the fitness of each particle.
3. Then, compare each particle's personal best to its present fitness, and set *si* to the greater performance, i.e.

$$s_i^{(t)} = \begin{cases} s_i^{(t-1)} & \text{if } f\left(p_i^{(t)}\right) \leqslant f\left(s_i^{(t-1)}\right) \\ p_i^{(t)} & \text{if } f\left(p_i^{(t)}\right) > f\left(s_i^{(t-1)}\right) \end{cases} \quad (5)$$

4. Put the global best $\widehat{\boldsymbol{S}}^{(t)}$ to the particle in the swarm with the best fitness inside the swarm, e.g.

$$\hat{s}^{(t)} \in \left\{s_1^{(t)}, s_2^{(t)}, \dots, s_m^{(t)}\right\} \mid F\left(\hat{s}^{(t)}\right) = max\left\{F\left(s_1^{(t)}\right), F\left(s_2^{(t)}\right), \dots, F\left(s_m^{(t)}\right)\right\} \quad (6)$$

5. According to Eq (5), change the particle's velocity vector (9). Each particle must be moved in accordance with Eq (5).
6. Assume that t Equals t plus 1.
7. Then, go back to step 2 and repeat until the stop condition has been reached.

When using PSO to solve optimization issues, there are two crucial steps: the representation of the solution and the fitness function. PSO's ability to use real numbers as particles is one of its best features. For example, it's not a genetic algorithm like [24] , which requires binary encoding transformation and particular genetic operators to work. As well as the way to accomplish picture segmentation, the comprehensive application of PSO will be presented in the next part.

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

.

4. The proposed IoT botnet detection method

There are two key steps to our proposed anomaly detection approach, as shown in Figure 1. (1) pre-processing of data and (2) proposed method are shown. The botnet problem is explained in the first section of this document. Next, the phases of the suggested method are described, which include normalisation, cleaning, and integration of data, among others. As a next step, ONE-SVM hyperparameters are found using PSO, and then feature selection is performed at each specific device in the optimization set. All phases of the suggested methodology are described at length in the following paragraphs.

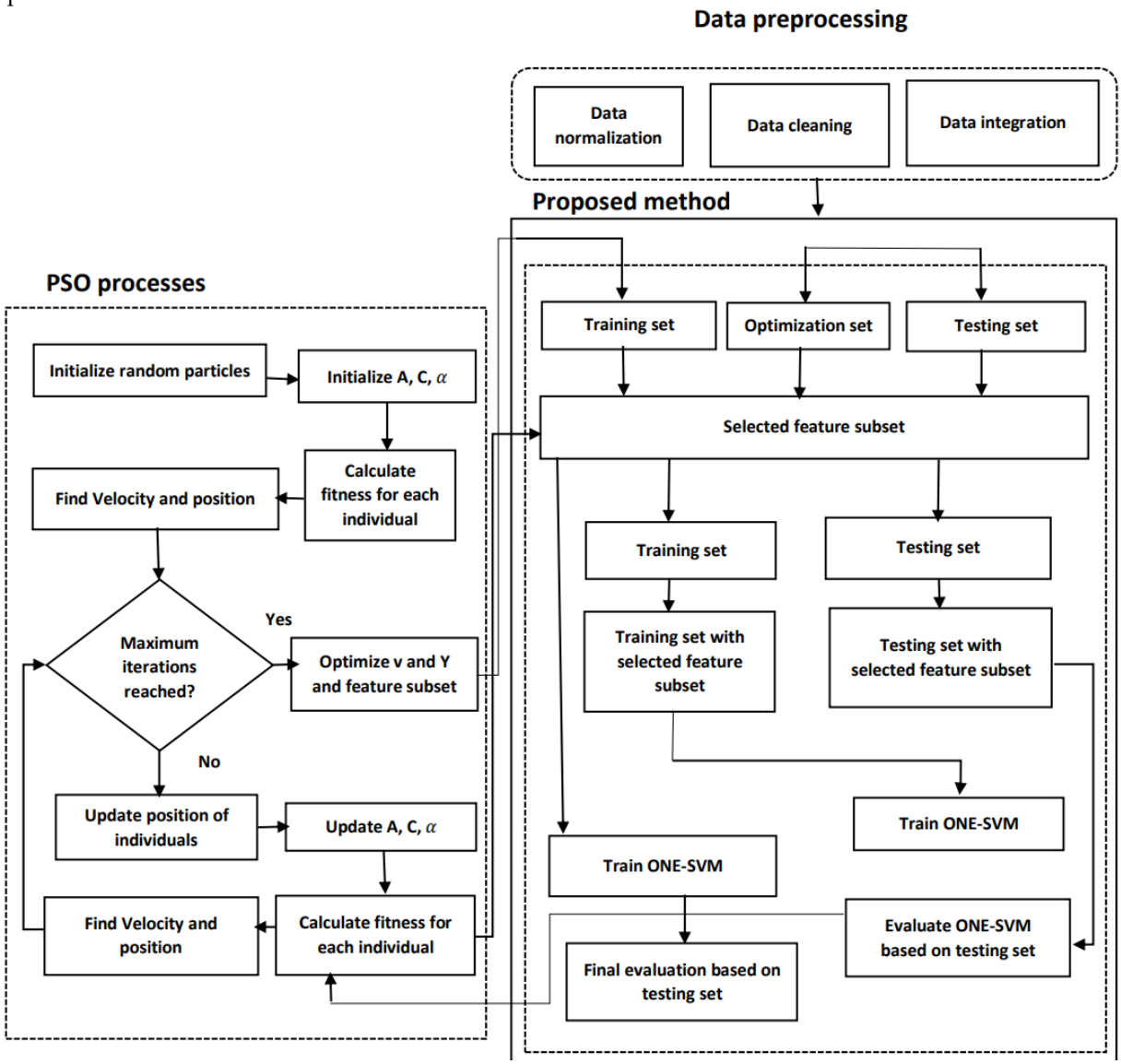

Figure 1: The propose model stages for detecting botnets in IoT.

4.1 Problem definition

Because it automatically disconnects compromised IoT devices from the network to prevent the spread of these attacks, instantaneous botnet detection can help improve IoT security. IoT botnet detection

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

approaches have traditionally relied on supervised learning, which has several drawbacks. A huge amount of storage space is needed to store a large number of examples of each type of malicious traffic, along with an equal number of benign traffic. As a second step, an expert is required to determine whether a packet is malicious or not. As a result, this will take a lot of time. A high number of IoT botnet attacks are conducted every day, making this technology ineffective at detecting them. There is an increasing need for unsupervised and faster approaches to detect IoT botnets as a result of these three issues. The goal of this work is to design an evolutionary network-based approach to tackle these difficult detecting challenges.

4.2 Proposed methodology

4.2.1 Data Pre-processing

The overall quality of the final model and/or the time required for actual training may be affected if raw datasets are used without any preprocessing. This is why it is possible to apply a variety of prepossessing tactics [25]. Since the N-BaIoT dataset is fresh, large, and based on real-world data, it may have inconsistency and redundancy. Data normalization, Data cleansing, data integration, and data reduction are some of the preprocessing procedures used to make the most of it. Each of these strategies will be described in more detail in the subsections that follow them.

4.2.1.1 Data cleaning

As a pretreatment approach, data cleaning is critical in ensuring that a dataset appropriately reflects the problem being tackled. It seeks to reduce inconsistency from the dataset because inconsistency can influence the accuracy of the generated models. It is possible, as a result, to improve the overall quality of the final model. Using the N-BaIoT dataset, the number of packets from source MAC IP, source IP, channels and sockets are dispersed. It also has five other elements that represent the dispersion of the time between packet arrivals, as well. To calculate 15 of these features, standard deviation is used. Standard deviation is the square root of variance, hence it is possible to unify the way features are calculated by utilising standard deviation or variance as a unit of measurement. In this study, the variance is employed as a measure of statistical significance.

4.2.1.2 Data integration

In general, IoT devices, according to [18], are task-oriented devices. Integration of data for each device type is therefore a simple process. For each device type, this integration translates the requested functionality into a single typical traffic pattern. This can be improved both in accuracy for each device type and in speed. In addition, the number of created models is reduced as a result of the optimization. An updated N-BaIoT dataset is shown in Table 1, which was created by integrating data according to device type and the amount of packets consumed during training, optimization and testing. The new N-BaIoT dataset called NN-BaIoT was developed to test the proposed approach.

Table 1: Training, optimization, and testing steps of the NN-BaIoT dataset

Type of device	Training	Testing		Optimization	
	Benign objects number	Benign objects number	Malicious objects number	Benign objects number	Malicious objects number
Doorbell	26,128	26,128	866,324	26,128	442,453

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

Webcam	16,333	15,002	214,342	15,445	15,603
Baby Monitor	55,346	54,453	616,543	53,453	53,367
Security camera	68,243	68,134	2,234,156	68,490	1,054,342
Thermostat	4427	4472	534,445	4427	548,456

4.2.1.3 Data normalization

In the context of data mining, data normalization is crucial when working with a dataset that contains features of varying scales. In order for all features to contribute equally to the model, the scale must be uniform. The NN-BaIoT dataset only contains numeric features. Using the Min-Max normalization described in Equation, all of them can be normalised (10). As a result, all features in the NN-BaIoT dataset have been rescaled to reside in the [0, 1] range.

$$A_i = \frac{A_{oi} - minA_{oi}}{maxA_{oi} - minA_{oi}} \quad (7)$$

When normalizing a dataset, A_{oi} represents the i^{th} instance, while Ai represents the i^{th} instance after normalization. Also, **n** represents how many instances are in the supplied dataset, and *i* represents an integer number from 1 to n, respectively.

4.2.1.4 Data reduction

A real-world dataset may contain aspects that are redundant and irrelevant. By reducing the number of superfluous features, data reduction strategies can increase efficiency and classification accuracy. Feature selection is one of the most important data reduction methods. A predetermined evaluation metric is used to find the optimal subset of features. As a result of PSO algorithm, the feature subset for ONE-SVM has been optimized for ONE-SVM. Following is a description of this technique.

4.2.2 Proposed Method

This section explains how to identify IoT botnet assaults using PSO and ONE-SVM. Where PSO is used to optimize the hyper-parameters of ONE-SVM and to perform feature selection at the same time. Wrapper feature selection strategy is used instead of filter approach because it is more significant than filter approach [26]. There are three primary components to a classification task: a learning algorithm, a search algorithm, and an evaluation measurement [27]. In order to evaluate the selected subset of features, the wrapper feature selection strategy incorporates the entire learning process. ONE-SVM is employed as a learning algorithm and FPR(1-TPR) is used as an assessment metric in the design of the proposed approach. IoT botnet problem was solved using PSO and ONE-SVM algorithm. As a result, the following two critical points should be addressed:

- Individual representation: in order to address this issue, it is necessary to determine the decision factors that reflect individuals. They include ONE-SVM hyperparameters and all features from the NN-BaIoT dataset. An individual is therefore expressed as a 1-dimensional array of real numbers comprising 117 items, according to Eq (8). The last two components of the array are the potential hyperparameters of ONE-SVM v and the kernel parameter $\boldsymbol{\lambda}$. In other array elements, boolean variables are represented, with each variable representing a feature. So, if the boolean variable's value is greater than or equal to 0.55, the feature is picked, else it is ignored.

$$I_i = [R_1, R_2, \dots R_{115}, v, \gamma] \quad (8)$$

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

- It is necessary to evaluate an individual's quality based on a set measurement to address this point. Accordingly, the fitness function should be chosen based on the problem at hand, and vice versa.
- IoT botnet's fitness function is estimated using an optimized one-SVM model to minimize FPR and 1-TPR using the proposed method. In this case, the ONE-SVM model is trained by using the ONE-SVM parameters and the features denoted by the person. Individual I at iteration t will be evaluated using the fitness function described in Eq. (9) to determine the quality of the individual. So that overfitting can be avoided, the fitness function is evaluated using the NN-BaIoT dataset optimization set. This results in more resilient models.

$$\text{Fitness}\left(I_i^t\right) = FPR_i^t\left(1 - TPR_i^t\right) \quad (9)$$

PSO-ONE-SVM algorithm's essential steps are explained below once the two design issues have been addressed:

1. Initialization: Starting with random initialization of the first population, the PSO-ONE-SVM model uses ONE-SVM parameters (v and λ) and various variables to represent the features in the dataset (see Eq. 8).
2. Evaluating each individual is done by using the fitness function that was previously given in Eq. (9).
3. Check for updates to search agents PSO's reproduction operators are used to create a new population in this step. This means that agents' locations are updated using the PSO updating equations, which ensure a balance between exploration and extraction of resources.
4. Termination: Whenever the specified maximum number of iterations has been reached, a search for the optimum solution will come to a halt. The PSO-ONE-SVM model now provides the best hyperparameters of ONE-SVM and the best subset of features with the highest fitness score.
5. The created models are assessed based on a testing component in this step. This step's evaluation measures are described at the next section.

5. Experiments and results

Through trials on the NN-BaIoT dataset, the suggested PSO-ONE-SVM approach is evaluated. The performance of the suggested approach is compared to other algorithms typically used for anomaly detection, such as OCSVM, GWO-OCSVM and IF for verification [28]. These sections describe and analyze the experiments and their outcomes in depth.

5.1 Evaluation measures

For the proposed IoT botnet detection approach, the performance is evaluated using the following three evaluation measures: (1) True Positivity (TPR). In other words, it is the ratio of true positives to the total number of true positives and false negatives:

$$TPR = \frac{TP}{TP + FN}$$

(2) False Positive Rate (FPR) is the ratio of false positives to the sum of false positives and true negatives:

$$FPR = \frac{FP}{FP + TN}$$

(3) Indicator of geometric-mean (G-mean) Essentially, it is the product of the square root of TPR and 1 – FPR

$$G - \text{mean} = \sqrt{TPR(1 - FPR)}$$

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

The testing set of the dataset is used to determine the TPR, FPR, and G-mean. Greater FPR, G-mean, and TPR values suggest stronger anomaly detection models, with the perfect model occurring when FPR, G-mean, and TPR are equal to 1.

5.2 Experiments environment and setup

To build this approach and other anomaly detection algorithms, Anaconda Python framework version 5.3 is employed. EvoloPy [13] uses the open-source code of PSO to implement the swarm optimization method. EvoloPy is a Python-based open-source optimization framework. Metaheuristic algorithms are included in the framework. As demonstrated by the designers of EvoloPy, the Python implementation of PSO has a faster running time than its Matlab counterpart for large-scale problems. In all trials, a Windows Server 2012 64-bit OS and an Intel(R) Xeon(R) CPU E5 2609 with 64 GB RAM were used as the operating system platforms. Of the benign, 1/3 of the data from each of the five IoT device types were used as a training, optimization, and testing approach (i.e., the training set of each device type). These experiments were carried out to better understand the normal patterns of traffic on the Internet. PSO-ONE-SVM hyperparameters were optimised and the optimal features subset was determined by training and testing each IoT device type with 1/6 benign data and 1/3 malicious. A device's test data is made up of 1/3 benign data and 2/3 harmful data. Using grid search for hyperparameter tuning, the other three algorithms may be compared fairly.

When using PSO-ONE-SVM, the number of iterations is set to 20. The PSO-ONE-SVM method converges to this figure after numerous initial experiments. Currently, the number of participants is set at 26. Search agents are limited to 7. According to well-regarded publications, PSO's additional parameters are set. Due to the normalization of the dataset, the lower bound ***lb*** and upper bound ***ub*** are both set to 0. For example, OCSVM, GWO-OCSVM and IF were all employed to detect anomalies in this paper. Table 2 lists the optimized hyperparameters for these algorithms. Using PSO, PSO-hyperparameters ONE-SVM's are optimized, whereas grid search is used to optimize the hyperparameters of other methods.

Table 2: Settings of parameter

Algorithm	**Parameters**	**Selected values**				
		Thermostat	Webcam	Baby monitor	Doorbell	Security camera
PSO-ONE-SVM	V	0.053	0.009	0.081	0.024	0.046
	Y	0.220	0.051	0.014	0.025	0.01
	Kernel	R-B-F	R-B-F	R-B-F	R-B-F	R-B-F
OCSVM	V	0.003	0.006	0.0022	0.004	0.015
	Y	0.044	0.053	0.008	0.01	0.084
	Kernel	R-B-F	R-B-F	R-B-F	R-B-F	R-B-F
IF	Behaviour	N-E-W	N-E-W	N-E-W	N-E-W	N-E-W
	Cont_amination	0.037	0.014	0.025	0.011	0.05
	Estimators_n	220	220	220	220	220
	maxSamples	1023	1023	1023	1023	1023
GWO-OCSVM	V	0.066	0.008	0.055	0.043	0.04
	Y	0.230	0.045	0.010	0.016	0.008

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

5.3 Dataset description

A benchmark dataset, N-BaIoT [18], was downloaded from the University of California, Irvine (UCI) machine learning repository [29]. The N-BaIoT dataset is studied and used in this work. Numerous aspects make this dataset stand out from previous IoT botnet datasets. Here are a few examples:

- A semi-online technique can be applied to the N-BaIoT dataset because the derived features are incremental statistics (gather instances in batch to train a model and then discard). There are no storage issues, hence the proposed model's training is practical.
- In most cases, botnets are built and controlled in various stages, such as bot infection, propagation, connection with C&C server, and conducting various types of malicious actions [5]. It is insufficient to merely determine the early phases of infection according to[18]. For example, in the N-BaIoT data set, the last stage of botnet formation is when the IoT bots start conducting assaults.

The use of this technology, therefore, adds a last line of defence to the proposed strategy. This is the first true dataset that has been used in the literature to detect IoT botnet assaults, as far as we are aware.

Large organisation networks, which are likely to see an increase in the number of Devices, are used to collect data for the N-BaIoT datasets.

As well as normal and attack situations, this collection contains statistics on the network traffic of an IoT system. On the IoT network, we have installed a baby monitor and two doorbells as well as four security cameras and a thermostat. Wi-Fi is commonly used to connect these devices to IoT networks. We collected data on every single device both in its default operating mode and when attacked by the BASHLITE and Mirai botnet.

It is possible to forget information over time if you have 23 recorded features in Table 1 and 5 values of the factor $\boldsymbol{\lambda}$ =(0.01, 0.10, 1, 3, 5) utilised in-stream clustering This means that there are 24.2, multiplied by five, or 115 characteristics.

5.4 Results and discussion

It is shown in Table 3 how the proposed method and other anomaly detection algorithms were performed in testing on the NN-BaIoT dataset. Note that greater TPR values (detection of attacks after they occur) and lower FPR levels (misclassification of benign data as malicious) equate to better results. Figure 2, 3 and 4 are charts that illustrates the relative performance of the proposed algorithm in comparison to previous anomaly detection algorithms. As you can see in this chart the suggested algorithm's TPR and FPR values are significantly higher than those of the other anomaly algorithms, indicating that the hyperparameters and feature subset identified by PSO-ONE-SVM are likely to be the most effective. OCSVM, on the other hand, has a lower FPR than the IF, but a lower TPR than OCSVM. IoT devices were found to have a highly variable TPR and a very variable FPR using OCSVM and LOF, respectively. Thermostat, webcam, doorbell, and security camera are examples of IoT devices with limited and deterministic capabilities according to [18]. Other IoT devices, such as a baby monitor, have a wide range of functions. Note that the suggested approach outperformed OCSVM in terms of TPR and FPR, while the deep autoencoder was unable to capture the regular traffic pattern for baby monitor devices, as described by [18].

Table 3: NN-BaIoT dataset's FPR and TPR

Device type	GWO-OCSVM		OCSVM		IF		PSO-ONE-SVM	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Thermostat	0.960	0.009	0.498	0.139	0.992	0.101	0.996	**0.007**
Webcam	0.999	0.037	0.999	0.186	0.999	0.692	**0.999**	0.025
Baby monitor	0.991	0.016	0.234	0.001	0.618	0.003	0.995	**0.001**

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

Doorbell	0.995	0.083	0.923	0.003	0.826	0.010	**0.998**	0.083
Security camera	0.982	0.098	0.813	0.039	0.999	0.419	**0.999**	**0.03**

Because IoT devices tend to be task-oriented, their restricted capability can be translated into a limited number of conventional traffic patterns. Because the NN-BaIoT dataset is unbalanced, the G-mean is also

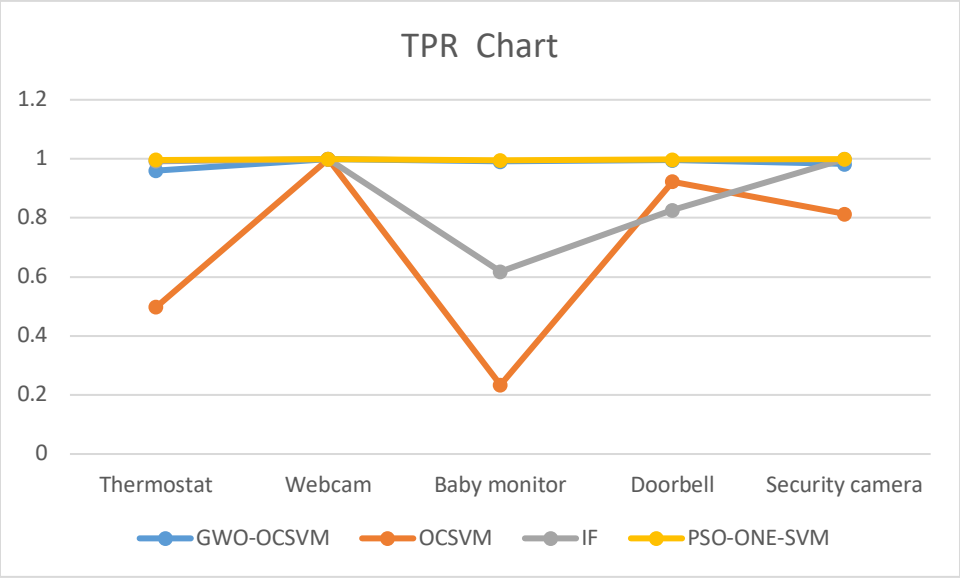

Figure 2: TPR for the proposed method compared with previous methods.

employed for the imbalance. Table 4 shows the classification results for the NN-BaIoT dataset in terms of the G-mean metric at the testing stage Because its average among IoT devices is 0.988, whilst the other algorithms are all below 0.7, the suggested algorithm surpasses the others. A suitable balance between TPR and FPR values can only be achieved by using the algorithm that has been provided here. According to Table 5, existing anomaly detection techniques are compared with the proposed approach in terms of their average detection time over the NN-BaIoT dataset at the testing stage. The suggested approach is superior to the previous three anomaly detection algorithms, requiring an average of only 5 seconds to detect the attacks. The average DDoS assault lasts between 30 and 100 seconds, with 10% of attacks lasting more than a day and 3% lasting more than a month [30]. Considering that hacked IoT devices are immediately disconnected from the network once the attacks are started, these attacks can be stopped in less than five seconds. This is a significant reduction in DT.

Table 4: NN-BaIoT dataset's G-Mean

Device Type	G-mean			
	GWO-OCSVM	OCSVM	IF	PSO-ONE-SVM
Thermostat	0.975	0.655	0.944	0.**985**
Webcam	0.981	0.902	0.555	**0.989**
Baby Monitor	0.987	0.479	0.785	0.**995**
Doorbell	0.956	0.959	0.904	0.974
Security camera	0.941	0.884	0.762	0.963
Average	0.968	0.776	0.789	**0.987**

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

Table 5: Detection time on average

Device Type	Detection time			
	GWO-OCSVM	OCSVM	IF	PSO-ONE-SVM
Thermostat	0.047	0.087	3.325	**0.33**
Webcam	0.156	0.406	11.589	**0.121**
Baby monitor	0.622	1.889	49.698	**0.342**
Doorbell	1.099	1.57	18.208	1.002
Security camera	27.837	36.549	42.398	25.346

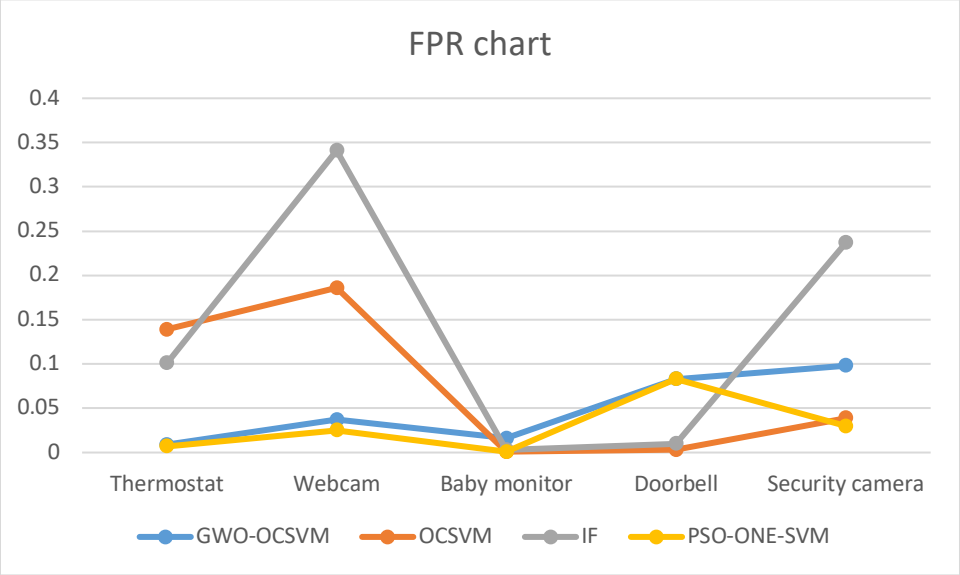

Figure 3: FPR for the proposed method compared with previous methods.

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

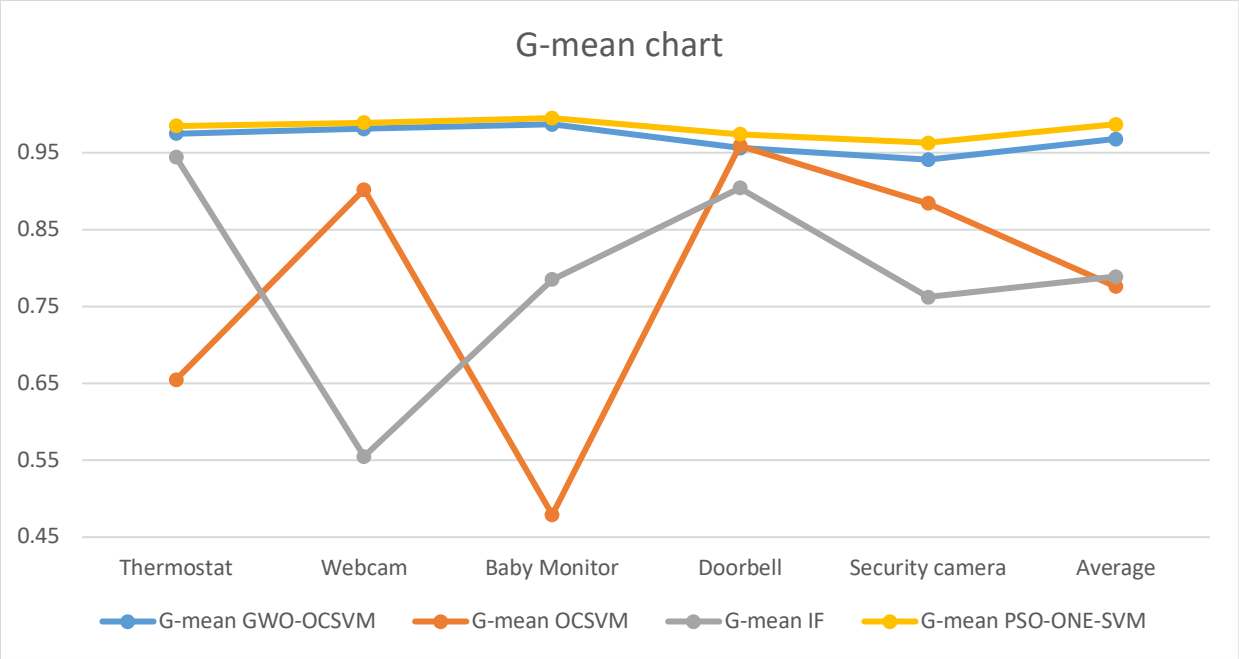

Figure 4: G-Mean for the proposed method compared with previous methods.

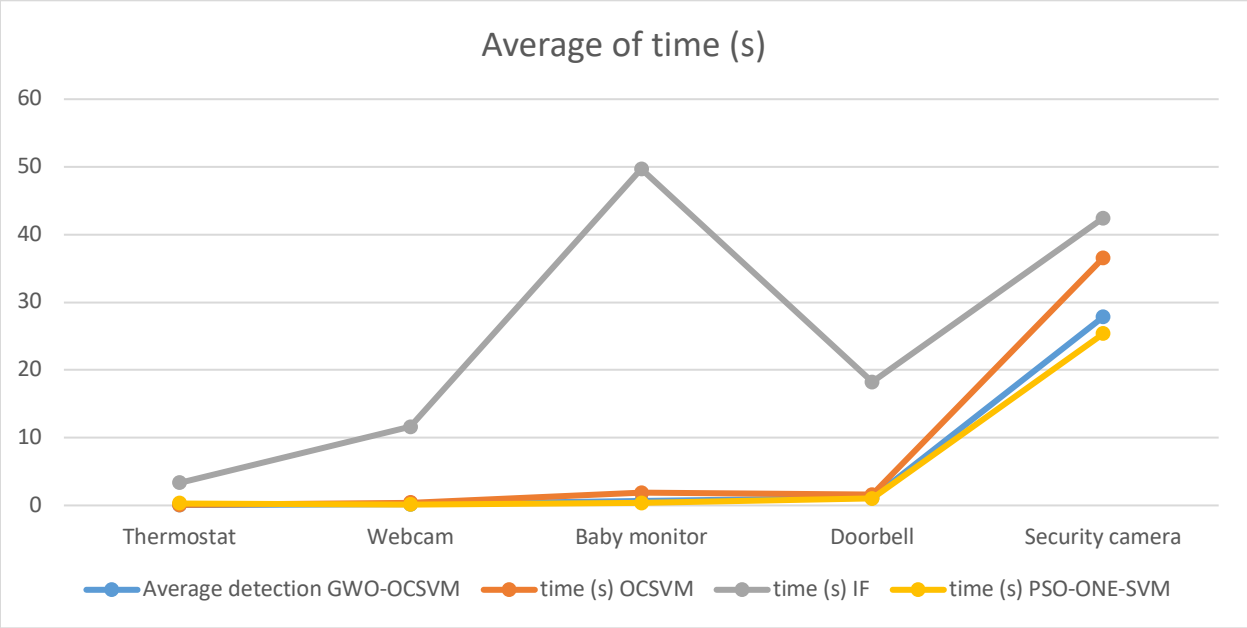

Figure 5: The average detection time for the proposed method compared with previous methods.

6. Conclusions and future works

This is due to the fact that the number of deployed IoT devices has exploded, they lack security, and the hacked IoT devices may not show any obvious indicators of infection. By utilising GWO to improve the hyperparameters of OCSVM and performing feature selection, the proposed

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

algorithm in this research aims to identify IoT botnet assaults by utilising GWO. The OCSVM classifier's performance was significantly improved by using the GWO on NN-BaIoT data. According to G-mean for all types of IoT devices, the proposed method surpasses the three previous unsupervised algorithms typically used for anomaly detection In addition, it reduces the number of selected features while achieving the lowest detection time. In the future, the suggested technique will be tested for different types of IoT devices using Big Data training data.

References

[1] K. Angrishi, "Turning internet of things (iot) into internet of vulnerabilities (iov): Iot botnets," *arXiv Prepr. arXiv1702.03681*, 2017.

[2] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 261–274, 2015.

[3] D. Celebucki, M. A. Lin, and S. Graham, "A security evaluation of popular internet of things protocols for manufacturers," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1–6.

[4] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Futur. Gener. Comput. Syst.*, vol. 82, pp. 395–411, 2018.

[5] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer (Long. Beach. Calif).*, vol. 50, no. 7, pp. 80–84, 2017.

[6] S. Mansfield-Devine, "DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare," *Netw. Secur.*, vol. 2016, no. 11, pp. 7–13, 2016.

[7] E. Bertino and N. Islam, "Botnets and internet of things security," *Computer (Long. Beach. Calif).*, vol. 50, no. 2, pp. 76–79, 2017.

[8] H. Wang, J. Gu, and S. Wang, "An effective intrusion detection framework based on SVM with feature augmentation," *Knowledge-Based Syst.*, vol. 136, pp. 130–139, 2017.

[9] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[10] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemom. Intell. Lab. Syst.*, vol. 149, pp. 153–165, 2015.

[11] X. Zou, J. Cao, Q. Guo, and T. Wen, "A novel network security algorithm based on improved support vector machine from smart city perspective," *Comput. Electr. Eng.*, vol. 65, pp. 67–78, 2018.

[12] M. H. Nguyen and F. la Torre, "Optimal feature selection for support vector machines," *Pattern Recognit.*, vol. 43, no. 3, pp. 584–591, 2010.

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

[13] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018.

[14] R. Hallman, J. Bryan, G. Palavicini, J. Divita, and J. Romero-Mariona, "IoDDoS-the internet of distributed denial of sevice attacks," in *2nd international conference on internet of things, big data and security. SCITEPRESS*, 2017, pp. 47–58.

[15] S. Garc\'\ia, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," *Secur. Commun. Networks*, vol. 7, no. 5, pp. 878–903, 2014.

[16] K.-C. Lin, S.-Y. Chen, and J. C. Hung, "Botnet detection using support vector machines with artificial fish swarm algorithm," *J. Appl. Math.*, vol. 2014, 2014.

[17] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the rise of IoT compromises," 2015.

[18] Y. Meidan *et al.*, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, 2018.

[19] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv Prepr. arXiv1802.09089*, 2018.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[21] N. M. Hatta, A. M. Zain, R. Sallehuddin, Z. Shayfull, and Y. Yusoff, "Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014--2017)," *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2651–2683, 2019.

[22] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 7, pp. 2809–2825, 2020.

[23] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, 1995, vol. 4, pp. 1942–1948.

[24] L. Davis, "Handbook of genetic algorithms," 1991.

[25] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[26] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, and J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection," *Futur. Gener. Comput. Syst.*, vol. 55, pp. 376–390, 2016.

[27] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1–2, pp. 273–324, 1997.

[28] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognit.*, vol. 74, pp. 406–421, 2018.

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021

[29] T. E. Dheeru D, "UCI machine learning repository.," 2017. http://archive.ics.uci.edu/ml.

[30] N. Blenn, V. Ghiëtte, and C. Doerr, "Quantifying the spectrum of denial-of-service attacks through internet backscatter," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–10.

DOI: 10.5281/zenodo.5203167
Received: April 13, 2021 Accepted: July 19, 2021