Personal and Ubiquitous Computing (2024) 28:123–133
https://doi.org/10.1007/s00779-021-01567-8

ORIGINAL ARTICLE

Soft computing for anomaly detection and prediction to mitigate IoT-based real-time abuse

M. P. S. Bhatia[1] · Saurabh Raj Sangwan[1]

Received: 1 March 2021 / Accepted: 15 April 2021 / Published online: 10 May 2021

Abstract

Cyber-surveillance and connected devices can be misused to monitor, harass, isolate, and otherwise, harm individuals. In particular, these devices gather high volumes of personal data such as account details with shared passwords, person's behavior and preferences, movements by GPS, and audio-video recordings which can be maneuvered. It is therefore imperative to define approaches that help mitigate the Internet of things (IoT)-based real-time abuse in a pro-active, reactive, or predictive manner. The key objective of this research is to outline and categorize such approaches. Further, to comprehend predictive analytics as a potential solution to mitigate technology abuse, we propose an anomaly detection methodology (MF_EW_Bagging) to classify normal and abnormal use pattern categories in an Intrusion Detection System (IDS) for IoT system. A hybrid feature selection technique based on an *ensemble* of *multiple filter*–based techniques and a *wrapper* algorithm is firstly used as search method for finding an optimal feature subset. Further, ensemble learning technique, namely *bagging*, is used for final classification into normal and abnormal use pattern categories. The use of ensemble feature selection removes biasness of individual feature selection method during ensemble and identifies the optimal subset with non-redundant and relevant features. The proposed methodology is evaluated on publicly available real-time IDS dataset. The research persuades the need of designing robust and lightweight IDS for IoT-based smart environments which understand the cyber-security risks in a proactive predictive manner as it the best way to defend networks and systems with the growing IoT complexity.

Keywords Security · Soft computing · Abuse · IoT · Anomaly · Intrusion

1 Introduction

According to the World Health Organization, interpersonal violence is a leading cause of impaired quality of life and mortality in the world, especially among people between 15 and 44 years [1]. Interpersonal abuse and violence is a pattern of behavior used to establish power and control over another person through fear and intimidation, often including the threat or use of violence [2]. It can take many forms such as verbal or emotional, physical, sexual, and digital abuse; stalking (online and in-person); and economic abuse which can put public security at risk and cause detrimental effects on the personality and esteem of the individual. Physical assaults, stalking, and verbal insults occur in real time but are often unreported or suffer delayed response. A person can be abused physically, sexually, or verbally in-person in a real-time environment. It is imperative to record and detect such acts and their severity. Internet of things (IoT) is one of the most disruptive technologies to surface in modern history and can potentially assist in apprehending instances of real-time abuse (RTA). The IoT devices such as smart wearable devices have emerged as a powerful monitoring system for detecting psychological state changes that might be triggered due to physical harm or fighting [3–5]. Also, surveillance systems can be used to tap vaping, substance abuse, real-time bullying, molestation, and cyber-stalking incidents. For example, sensors that capture elevated sound levels can facilitate detection of fighting or bullying. At the same time, technology means that abuse is no longer limited to schoolyards, street corners, or public places. Cyberspace has been recognized as a conducive environment for use of various hostile, direct, and indirect behavioral tactics to target individuals or groups [6–8]. An alarming trend in the abuse of everyday technology has been recently observed where perpetrators misuse networked

✉ Saurabh Raj Sangwan
saurabh.trf18@nsut.ac.in

[1] Department of Computer Science & Engineering, Netaji Subhas University of Technology, New Delhi, India

devices and software to control, isolate, humiliate, and dominate their victims. Collectively referred to as technological abuse or technology-facilitated abuse, it can range from online harassment, stolen online identities, hacking, spoofing, and revenge pornography to stalking and surveillance.

With the vast benefits that the connected world brings to the consumers, it is also inviting attackers to continuously identify new exploits and hit techniques designed to circumvent the security around the IoT networks. IoT is connected to the worldwide Internet that exposes it to global intrusion in addition to wireless attacks inside an IoT network. That is, though the networked devices provide many advantages, they also offer abusers an abundance of opportunities to control, harass, and stalk their victims. Unfortunately, IoT-based surveillance and real-time data can itself create an abusive environment or cyber-stalking incident. The IoT devices collect huge amount of data at the granular level including account details with shared passwords, person's behavior and preferences, movements by GPS, and audio-video recordings. The ubiquitous sensing enabled through these IoT devices exacerbates the forms of abuse. For example, a set of cameras that were being accessed from the same remote location dozens of times can raise an anomaly as it may turn out that an employee of a service provider is improperly using the cameras to watch people in their homes. Users can take measures such as changing network settings or replacing the risked smart devices. But, such tactics will turn futile if a perpetrator uses stalkerware (software designed to monitor messages on a device, record screen activity, track its location, and give access to its cameras) or has access to an Internet router and realizes a password change.

Network forensics [9] and penetration testing (ethical hacking) [10] are primary methods which can be used to identify security vulnerabilities. The generic term "forensic" involves the use of scientific methods and techniques to investigate a crime. This term can be aptly adapted and applied in the same way within the context of networks. Referred to as network forensics, it involves the use of scientific methods and techniques to capture, store, and analyze network events and attacks. Techniques include detection, which implies the ability to detect the presence of an attack as early as possible, and prediction, which means deriving the likelihood of future attacks from current data. Figure 1 depicts primary tasks for improving security systems and preventing attacks. But, the current cyber-security solutions leave a wide gap in coverage as the variety and volume of data involved in identifying and predicting security threats are overwhelming.

There are a plethora of tools and solutions available to detect attacks and block cyber-attacks such as firewalls, spam filters, and anti-malware to protect endpoints across environments (home or organizations), regardless of size or industry. However, another highly valuable security tool that is indispensable to ensure network security is the Intrusion Detection

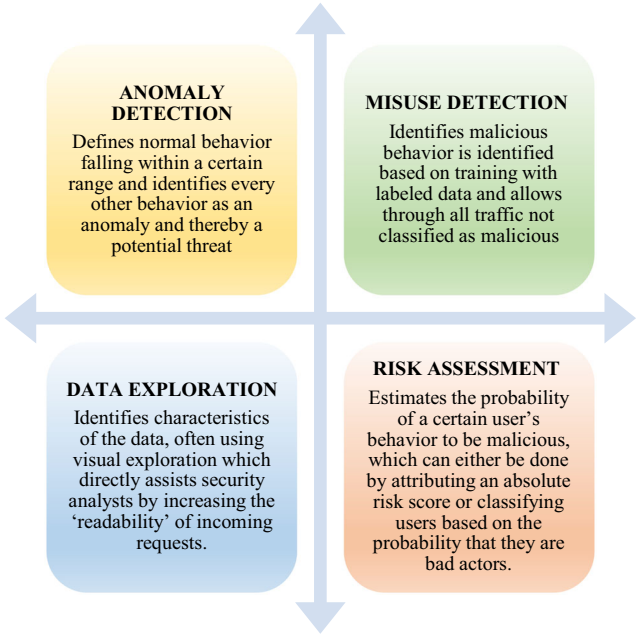

Fig. 1 Tasks for improving security systems and preventing attacks

System (IDS). IDS is a primary mechanism for cyber-security as data collected via networked devices can be analyzed for detecting illegal access and threats. Typically, an IDS is an application which deals with threats from the Internet or the Intranet [11, 12]. It is further categorized as signature-based IDS and anomaly-based IDS [13]. The signature-based approach include traditional techniques to fight cyber-attacks by gathering data about malware, data breaches, phishing campaigns, etc., and extracting relevant data into signatures, that is the digital fingerprint of the attack. These signatures are then compared against files and network traffic that flows in and out of a network in order to detect potential threats. While signature-based solutions continue to remain a prevalent form of protection, they do not suffice to deal with the advanced and increasingly sophisticated attacks. Since most attacks do not occur in the predefined pattern lists, signature-based techniques cannot protect the system against unknown attacks. Also as the types and frequency of attacks are growing continuously, it becomes time-consuming and impracticable to keep the database updated. In an anomaly-based IDS, the system traffic is tracked and correlated to the system's usual activities and usage. Some variation from the normal pattern of use is regarded as an interference. The anomaly detection method can identify novel attacks which have not been previously defined whereas signature-based detection can fail under such circumstances.

An IDS is a security mechanism that works mainly in the network layer of an IoT system. It is deployed for an IoT system which should be capable to analyze data packets and generate real-time responses, analyze data packets in different layers of the IoT network with different protocol stacks, and adapt to different technologies in the IoT environment [14].

Also, an IDS that is designed for IoT-based smart environments should operate under constrained conditions in terms of low processing capability, limited storage, battery, fast response, and high-volume data processing. The basic IDS categories for IoT are based on the detection, placement, and validation strategies adopted as shown in Fig. 2.

Predictive techniques for mitigating security issues can be used to defend against attacks, interference, and unauthorized access to information and computers. Recently, soft computing techniques have emerged with proven capabilities to analyze IoT device usage and behavior across very large deployments. A human observer would never be able to correlate the activities that signal abusive behavior. Intelligent machine learning techniques can facilitate monitoring access logs to a million security cameras and detect anomalies that might indicate abuse. This process is based on a software "agent," that is the IDS. IDS usually manages large magnitudes of data traffic and is challenged due to the dynamic, extensive, instant, and noisy data. Common issues include obtaining sufficient samples, redundant and inappropriate features, noise removal fallacy, and evaluation dilemma. In order to reduce the processing time and increase the IDS's performance, only the most relevant features are selected. Feature selection is an imperative step in any machine learning task, wherein a subset of most relevant features is selected from the entire feature set. It is the process of selecting an optimal subset of features with the aim of maximizing or minimizing an objective function. Previous studies confirm that selection of features allows narrowing down a subset of features, or attributes, to be used in the predictive modeling process, thereby reducing the computational cost of modeling and, in some cases, improving the performance of the model, too [15, 16]. From a taxonomic point of view, feature selection methods usually fall into one of the following four categories, namely filter, wrapper, embedded, and hybrid classes [17]. Feature selection techniques are advantageous as they can counter the curse of dimensionality, reduce the overall training time, curb overfitting, and increase model generalizability. Basically, the accuracy and generalization power can be leveraged by choosing a correct feature selection technique [18, 19]. But, selecting the important features without much loss of total information is a computationally extensive problem which manifolds when the huge percentage of data is unstructured and high-dimensional as the real-time network traffic big data. It is imperative to choose a feature selection technique to get insights about the features and their relative importance with the target variable.

Feature selection techniques are intended to reduce the number of input variables to those that are believed to be most useful to a model in order to predict the target variable. The feature selection techniques can further be divided into unsupervised and supervised techniques. The supervised feature selection techniques are further classified as filters, wrappers, and intrinsic techniques. The filter methods are based on the characteristic properties of the features such as *relevance* of the features which is measured via univariate statistics. In contrast to the filter methods, wrapper methods measure the *usefulness* of features based on the classifier performance. Using various swarm-based wrapper methods can eliminate the curse of dimensionality by removing unnecessary and improper features in the data. This research proffers the use of multiple filter methods, namely information-based (information gain) [20, 21], divergence-based (Relief-F) [22], and dependency-based (chi-square) [23] hybridized with swarm-based ant colony optimization (ACO) [24] wrapper methods to maximize the relevance and minimize the redundancy in feature set. Finally, the optimized feature set is used to train an ensemble learning model (bagging) and make the final predictions. Bagging reduces the variance while retaining the bias. The proposed anomaly detection methodology MF_EW_Bagging classifies use pattern into normal and abnormal categories. The proposed methodology is evaluated on publicly available real-time NSL-KDD IDS dataset [25].

The organization of paper is as follows: Section 2 characterizes the primary approaches to mitigate IoT-based real-time abuse followed by a brief overview of related work in Section 3. Section 4 discusses the proposed model, and Section 5 presents the results and discussion. Conclusion and future work is given in the last section (Section 6).

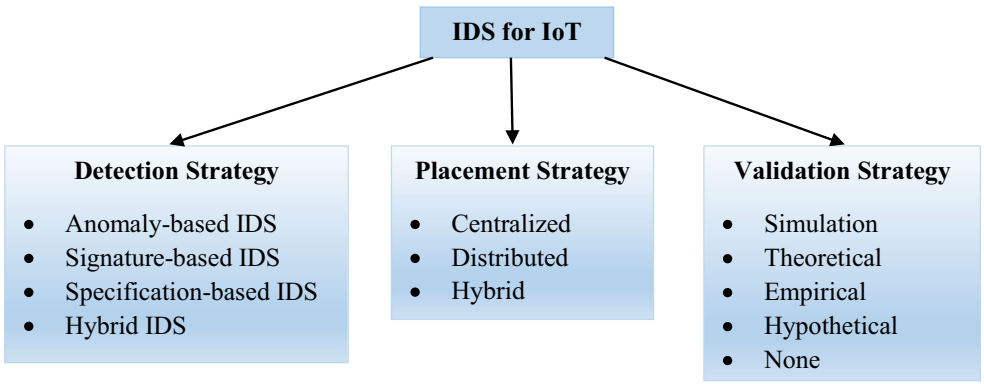

Fig. 2 IDS categories for IoT

2 Mitigating IoT-based RTA

IoT devices have dwelled expansively in our routine lives. Its pervasiveness as well as the intrusive data collection and sharing features transfigure these into digital weapons that can be used to harm, intimidate, and abuse people (children, adolescents, women, transgender) at various locations and in varied situational context. Moreover, the diverse data types and computing power among IoT devices mean there is no "one size fits all" cyber-security solution that can protect any IoT deployment. Therefore, it is imperative to outline approaches that help mitigate the IoT-based RTA in a pro-active or reactive manner. We categorize these approaches into four key types, namely overlooking the problem, prevention of IoT-based RTA, avoidance of IoT-based RTA, and finally, detection of IoT-based RTA. The categories proposed are homogeneous to the concept of deadlocks in operating systems. Prevention and avoidance are pro-active approaches whereas the detection is a reactive approach. With the recent upsurge in the use of learning-based techniques, we also look into a pro-active predictive category of mitigating IoT-based RTA. The following subsections illustrate these approaches.

2.1 Overlooking the problem

IoT implies that adequate devices are operational in a particular environment with dynamic communication. The IoT ecosystem enables information flows over the Internet with wireless accumulation and exchange of data. The growing numbers of IoT devices undoubtedly expand the capabilities of the environment but at the cost of a wider attack surface. Unfortunately, most users are unaware of the threats and vulnerabilities that may exist. At the same time, a majority of abuse in real time is ignored, owing to the non-willingness of victims to report such incidents. Ignorance of future risks and procrastination over taking action are never solutions, and providers need to mitigate cyber-security risk and build trust in the power of the IoT.

2.2 Prevention of IoT-based RTA

Preventing an abuse implies a situation when IoT-based abuse is bound to happen, but using some logic, we are preventing that abuse. Success depends on ensuring the integrity and confidentiality of IoT solutions and data while mitigating cyber-security risks. The following techniques can be used as preventive measures to mitigate the risks of IoT-based RTA:

- Changing passwords/passcodes for each account and device, including the Wi-Fi.
- Turning off GPS/location services/Bluetooth unless necessary.
- Preferable usage of a safe ("clean") device and a new account (email) which the abuser cannot access, for all safety planning.
- Remaining skeptical of suspicious messages, friend requests, emails, or attempts to collect user info from unknown third parties.
- Be careful with what you are posting, because it might give away information that would qualify as "social leakage."
- Keep all security apps/software updated.

2.3 Avoidance of IoT-based RTA

Avoidance refers to completely ruling out any chances of abuse. It is essential to stay ahead of the curve in order to avoid the detrimental consequences of compromised networks and faulty technology. This would mean considering some mandatory security guidelines as follows:

- *Data accountability*: All data being collected and stored within an IoT system should be accounted for.
- *Security settings*: All connected devices within a network should be configured with security in mind which includes setting strong username and passwords, multi-factor authentication, and encryption.
- *Device physical security*: It is important to physically safeguard IoT device against tampering. It should be kept in a restricted place or secured with the appropriate locks or other tools.
- *Life cycle approach for IoT security*: It is essential to adopt an end-to-end, comprehensive, policy-based architectural approach to address all the relevant security themes including network/application/hardware security, standards, detection and reaction, governance, and maintenance throughout the life cycle of an IoT object, that is from its manufacturing to disposal.

2.4 Detection of IoT-based RTA

Detection is essentially a reactive mechanism, where an abusive incident should be identified and reported promptly. Inherently, IoT devices have low computing power, custom architectures, and little memory and storage whereas the standard security solutions require some performance, are often hard to port to custom architectures, and require much memory and storage for database. Detection approaches look for identifying anomalies by mining insights or information in a data pool. Machine learning (ML) can offer a viable solution to compensate for this differential in settings as it looks for patterns in given data, is easy to port to new and unknown architectures, and requires little computing power, memory,

and storage. Pertinent studies report the use of artificial intelligence (AI) to detect anomalies by real-time modeling of network traffic, log and audit files, net nodes, servers, and all "smart IoT" devices. ML-based solutions can mitigate risks of new malware that can no defined "signature" (0-day attacks) and, at the same time, can counter the advanced persistent threats (APTs) where adaptive learning algorithms can detect the step-by-step penetration of apt malware (phishing, Trojans, adware, botnets, etc.).

Simultaneously, ML-based predictive analytics can be used to proactively detect and analyze threats, providing actionable insights to security analysts for making informed decisions with speed and accuracy. This research reports one such methodology where ML capabilities trained with optimal feature set are used to identify anomalies in real time such that observations detected and classified in the past can help to classify future data points. Prediction reduces the amount of time a security analyst may take to make the critical decisions and launch a systematic response to resolve the threat. Figure 3 depicts the effect of proactive vs. predictive vs. reactive mechanisms in mitigating IoT-based abuse.

3 Related work

Most of the existing literature discusses about the design of self-security devices, alarm systems, and SOS devices such as wearable, RFID tags, buttons, and GPS-GSM-enabled trackers, which would be used as reactive safety mechanisms during an abusive incident. Few studies also focus on *how* IoT devices can facilitate detection of abusive incidents and provide a pro-active mechanism. But to the best of our knowledge, none of the studies discusses solution-based approaches for IoT-based real-time abuse, that is how IoT can be misused for abuse. Simultaneously, most of the studies have focused on using IDS data for analyzing potential attacks. IDS has been a popular field of research for many years, and several systems for intrusion detection have been mentioned in the literature. Different researchers have reported the use of machine learning algorithms on various anomaly-based publicly available IDS datasets and evaluated performance [26–28]. In 2020, Chkirbene et al. [29] proposed a trust-based intrusion detection and classification system that limited input features' size based on a novel feature selection method. A model of network IDS based on convolutional neural network IDS was suggested by Xiao et al. [30] in 2019. In the same year, Kasongo et al. [31] proposed a deep learning–based IDS using neural networks combined with a feature selection algorithm based on filtering. A variety of filter and wrapper methods in anomaly detection systems have also been used in the literature. In 2016, Osanaiye et al. [32] had put forward an ensemble-based multi-filter feature selection (EMFFS) method that combines the output of four filters, namely information gain (IG), gain ratio, chi-square, and ReliefF to select important features. In 2020, Zhou et al. [33] proposed a heuristic algorithm with a voting classifier for intrusion detection and achieved an accuracy of 99.8%.

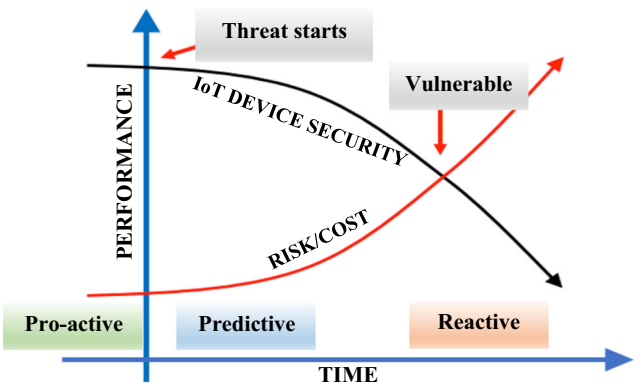

Fig. 3 Proactive vs. predictive vs. reactive mechanisms

4 MF_EW_Bagging: hybrid feature selection with ensemble learning to detect and predict anomalies for IoT-based RTA

One of the desirable characteristics of a ML model is that it should exhibit low variance, that is it should not overfit the training data and lose the generalization capabilities to unseen data. A key method is to minimize the number of features used to train the model. Feature selection techniques enable selecting a near-optimal set of input variables that would minimize variance and maximize generalizability of the model. These techniques optimize the model performance, reduce the training time, as well as make debugging and explainability easier with fewer features. Also, a single feature selection method may produce an optimal or sub-optimal local subset of features for which efficiency might be compromised. An ensemble feature selection approach combines multiple feature subsets to select an appropriate subset of features using a feature ranking combination that increases the classification accuracy. This paper puts forward a predictive analytic approach to understand its capabilities for IoT-based abuse mitigation. A multiple filter ensemble with wrapper-based feature selection is used to generate an optimal feature set which is used to train an ensemble learning Bagging classifier to output the class categories as normal or anomaly. Figure 4 depicts the proposed MF_EW_Bagging methodology.

4.1 Filter methods

Filter methods are used for selecting the most significant features from the given feature set. The filter methods are based on the characteristic properties of the features such as relevance of the features which is measured via univariate statistics. A number of filter methods are available in the literature,

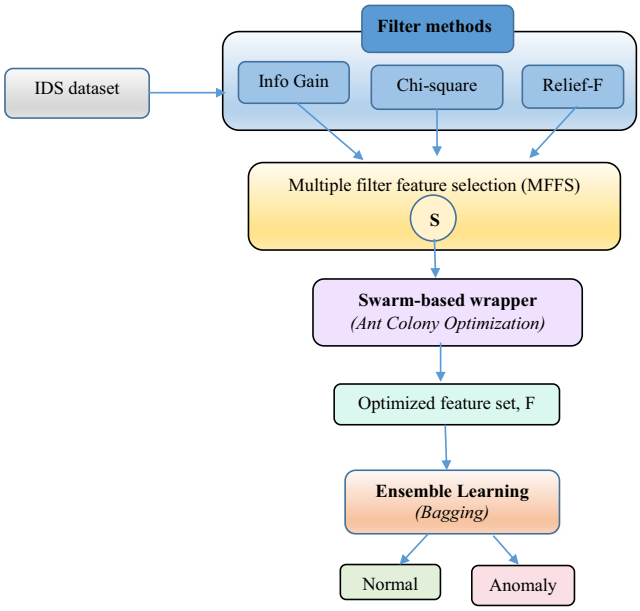

Fig. 4 Proposed MF_EW_Bagging methodology

broadly based on measures, like information (or uncertainty), distance, and dependence (or probability). In this work, we use the information-based information gain method, divergence-based ReliefF, and dependency-based chi-square method to typify a multi-filter which harnesses their combined strength and alleviate biasness on selected features.

4.1.1 Information gain

It is a method for calculating the relevancy of a particular feature for the determination of the class label. It measures the information gained in predicting a class value when a particular feature is present or absent. It is based on the concept of entropy and can be defined as "a measure of the reduction in entropy of the class variable after the value for the feature is observed." It can be calculated as given in (1).

$$\mathrm{IG}(t) = -\sum_{i=1}^{m} p(ci)\log p(ci) + p(t)\sum_{i=1}^{m} p(ci|t)\log p(ci|t) + p\left(t'\right)\sum_{i=1}^{m} p\left(ci|t'\prime\right)\log p\left(ci|t'\prime\right) \quad (1)$$

where c_i indicates the ith class; $p(c_i)$ indicates the probability of the ith class; $p(t)$ and $p(t')$ are the probabilities of the presence and absence of the feature t, respectively; and $p(c_i|t)$ and $p(c_i|t')$ are the conditional probabilities given the presence and absence of the feature t, respectively.

4.1.2 Chi-square

The chi-square (CS) test usually refers to Pearson's chi-square and is also known as the chi-square goodness-of-fit test or the chi-square test for independence. It is used when we have two categorical variables and want to determine whether there is a significant association between the two variables. It measures the dependence between stochastic variables, so using this function "weeds out" the features that are the most likely to be independent of class and therefore irrelevant for classification. It is calculated as given in (2).

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \quad (2)$$

where O_i is the observed value and E_i is the expected value.

4.1.3 Relief-F algorithm (Relief)

It is an instance-based, heuristic method; it works out weight values for each feature, based on how important they seem to be in discriminating between near neighbors. Algorithm 1 describes the working of the basic Relief filter method.

Algorithm 1: Relief Algorithm

Input: For each training instance a vector of attribute values and the class value
Output: Vector W of estimations of the qualities of attributes

1. *Set all weights W[A] := 0.0;*
2. ***For*** *i := 1* ***to*** *m* ***do begin***
3. *Randomly select an instance* R_i*;*
4. *Find nearest hit H and nearest miss M;*
5. ***For*** *A := 1* ***to*** *a* ***do***
6. $W[A] := \frac{W[A] - diff(A,R_i,H)/}{m} + \frac{diff(A,R_i,M)}{m}$;
7. ***End***

Relief-F evolved from the original Relief algorithm and was developed to improve its limitations. Kononenko [22] proposed a number of updates to Relief. Notably, the "F" in ReliefF refers to the sixth algorithm variation (from A to F) proposed. Firstly, ReliefF relies on a "number of neighbors" user parameter k that specifies the use of k-nearest hits and k-

nearest misses in the scoring update for each target instance (rather than a single hit and miss). This change increased weight estimate reliability, particularly in noisy problems. Secondly, three different strategies were proposed to handle incomplete data (i.e., missing data values). These strategies were proposed under the names Relief (B–D). Thirdly, two different strategies were proposed to handle multi-class end-points. These strategies were proposed under the names ReliefE and ReliefF. *ReliefF*, which inherited the changes proposed in ReliefA and ReliefD, was selected as the *best* approach. During scoring in multi-class problems, ReliefF finds *k*-nearest misses from *each* "other" class and averages the weight update based on the prior probability of each class (Algorithm 2).

Algorithm 2: ReliefF Algorithm

Input: For each training instance a vector of attribute values and the class value
Output: Vector W of estimations of the qualities of attributes

1. *Set all weights W[A] := 0.0;*
2. ***For*** *i := 1* ***to*** *m* ***do begin***
3. *Randomly select an instance* R_i*;*
4. *Find k-nearest hits* H_j*;;*
5. **For** all other classes (class C ≠ *class* R_i) ***do***
6. *From class C find k nearest misses Mj(C);*
7. ***For*** *A:= 1* ***to*** *a* ***do***
8. *Update weights:* $W[A] = W[A] - \frac{\sum_{j=1}^{k} diff\left(A,R_i,H_j\right)}{mk} + \frac{\sum_{C \neq class(R_i)} \frac{P[C]}{1-P[class(Ri)]} \sum_{j=1}^{k} diff\left(A,R_i,M_j(C)\right)}{mk}$;
9. ***End***

Next, a multiple filter feature selection (MFFS) technique is used to create a new search space which combines the best of all the three filter methods. That is, for the given feature set in the dataset, MFFS ranks and sorts the features according to the corresponding filter method. It then takes the top N features from each of the three filter rankings (R1, R2, and R3, respectively) and uses a union of set operation to include the best features from both the filter rankings, thus generating a selected feature set, S. Algorithm 3 depicts the working of the ensemble MFFS technique.

Algorithm 3: MFFS technique

Input: IDS dataset (NSL-KDD)
Output: Selected Feature Set (S)

1. IGRank = applyIG ( )
2. CSRank = applyCS ( )
3. ReliefFRank = applyReliefF ( )
4. R1 = Top N features IGRank
5. R2 = Top N features CSRank
6. R3 = Top N features ReliefFRank
7. S = (R1 ∪ R2 ∪ R3)

This feature set (S) may still be large, owing to the real-time dynamic data that is generated in large volumes and with high velocity, such as network traffic data. Therefore, the use of wrapper method is justifiable to find the most useful features.

4.2 Wrapper methods

In contrast to the filter methods, wrapper methods measure the usefulness of features based on the classifier performance. Given the large number of attributes, it is imperative to select the relevant few to shorten training time, enhance generalizability of the model by avoiding overfitting, get simplified models, and avoid the curse of dimensionality. Swarm algorithms are a class of population-based meta-heuristics which arrive at an optimum solution using a set of collective, decentralized, distributed, and self-organizing agents. The most prominent among the swarm-based algorithms are those inspired by the behavior of species in nature like birds, ants, and insects. In this paper, we use a swarm-based ACO wrapper algorithm as the search method for finding an optimal feature subset (F).

4.2.1 Ant colony optimization

Given by Dorigo [24] in 1992, it is inspired by the communication process used by ants. Ants, when searching for food, start off randomly in a direction. On finding food, the ant returns to its colony leaving pheromone (a chemical) trails on the way back. The pheromone is made stronger if other ants follow the path and find food as well. On the other hand, the trail becomes fainter as it evaporates over time if the path is not traveled by other ants. The pseudo-code for ACO is given in Fig. 5.

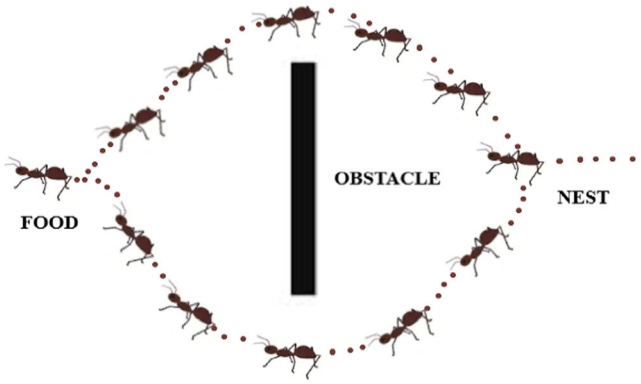

Fig. 5 Ant colony optimization

The algorithm for ACO is given in Algorithm 4.

Algorithm 4: Ant Colony Optimization

```
Begin
    Initialize pheromone and other parameters
    Generate a population of n ants
        for(ant i)
                Calculate fitness value
                Determine best position
    Determine best global ant (solution)
    Update pheromone trail
    Check stopping criterion
End
```

4.3 Ensemble classifier

Finally, ensemble learning is used to classify the intrusion. The ensemble classifiers combine the classification results from different classifiers to produce the final output. In this work, we use bagging. Bagging [34] refers to bootstrap aggregating which is a way to increase accuracy by decreasing variance. In bagging, each model in the ensemble votes with equal weight and trains each model with a random training set. It is done by generating additional dataset using combinations with repetitions to produce multi-sets of the same cardinality/size as original dataset.

5 Results and discussion

The performance of the proposed work through experiments is evaluated in this section. In this research work, we use the NSL-KDD IDS dataset [25]. This IDS classification was implemented on a 2.7-GHz Intel Core i5 with 16-GB RAM. NSL-KDD is an improved variant of the KDDCup 99 data collection that does not have redundant tests, avoiding a biased outcome for classifiers. It includes 41 features with class label attributes. The dataset has 41 features per record which can be further categorized in 4 types as shown in Fig. 6.

Table 1 enlists the features in the NSL-KDD dataset.

The metrics used to estimate the performance of the proposed work are given in Table 2.

In Table 2, true positive (TP) rate implies correctly classified anomalous instances as an anomaly, true negative (TN) rate implies correctly classified normal instances as normal, false negative (FN) rate implies wrongly classified anomalous instances as normal, and false positive (FP) rate implies wrongly classified normal instances as an anomaly. For a good classifier to detect attacks, it should have high DR and low false alarm rate (FAR).

The top N ranked features given by each filter were considered. The value of N was set to 14 as it divided the feature set into a 1/3 split. A union of set operation was then performed to generate a multiple filter feature set (S) with 19 features. These 19 features were input to the wrapper to output the most relevant and useful features, finally to generate the optimal feature set (F) which was used to train the ensemble classifier. The details of features selected using the individual filter methods, its ensemble, and subsequent wrapper are shown in Table 3.

The final selected 10 features using the multiple filter ensemble and the wrapper were as follows: f_3-Service, f_4-Flag,

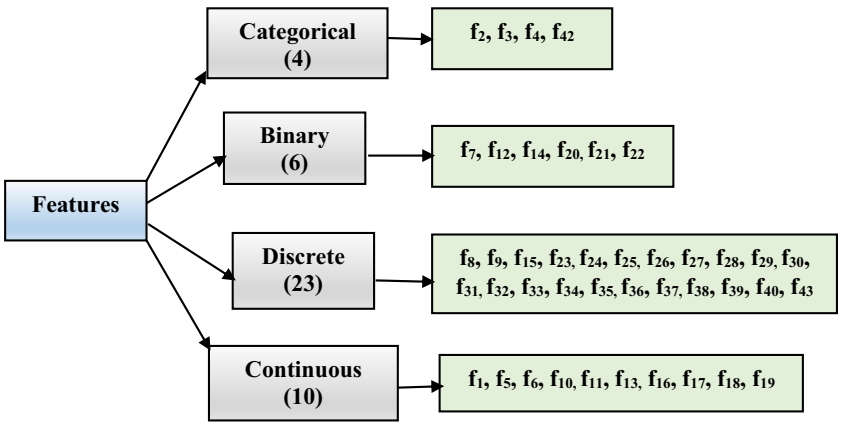

Fig. 6 Feature categories of the NSL-KDD IDS dataset

Table 1 Feature set of the NSL-KDD IDS dataset

Feature no.	Feature type	Feature name
f_1	Basic	Duration
f_2		Protocol_type
f_3		Service
f_4		Flag
f_5		Src_bytes
f_6		Dst_bytes
f_7		Land
f_8		Wrong_fragment
f_9		Urgent
f_{10}		Hot
f_{11}	Content	Num_failed_logins
f_{12}		Logged_in
f_{13}		Num_compromised
f_{14}		Root_shell
f_{15}		Su_attempted
f_{16}		Num_root
f_{17}		Num_file_creations
f_{18}		Num_shell
f_{19}		Num_access_files
f_{20}		Num_outbound_cmds
f_{21}		Is_hot_login
f_{22}		Is_guest_login
f_{23}	Traffic (same service features)	Count
f_{24}		Srv_count
f_{25}		Serror_rate
f_{26}		Srv_serror_rate
f_{27}		Rerror_rate
f_{28}		Srv_rerror_rate
f_{29}		Same_srv_rate
f_{30}		Diff_srv_rate
f_{31}		Srv_diff_host_rate
f_{32}	Traffic (same host features)	Dst_host_count
f_{33}		Dst_host_srv_count
f_{34}		Dst_host_same_srv_rate
f_{35}		Dst_host_diff_srv_rate
f_{36}		Dst_host_same_src_port_rate
f_{37}		Dst_host_srv_diff_host_rate
f_{38}		Dst_host_serror_rate
f_{39}		Dst_host_srv_serror_rate
f_{40}		Dst_host_rerror_rate
f_{41}		Dst_host_srv_rerror_rate

f_5-Src_bytes, f_6-Dst_bytes, f_{12}-Logged_in, f_{23}-Count, f_{26}-Srv_serror_rate, f_{29}-Same_srv_rate, f_{30}-Diff_srv_rate, and f_{39}-Dst_host_srv_serror_rate.

The performance results were evaluated for the proposed MF_EW_Bagging using accuracy, detection rate, and false alarm rate. The performance of individual filters, permutations of filter with wrapper, and wrapper was also evaluated. The

Table 2 Evaluation metrics

Metric	Explanation
Accuracy (ACC)	+ / + + +
Detection rate (DR)	/ +
False alarm rate (FAR)	/ +

Table 3 Features selected

Filter	No. of features selected	Ranked features
Information gain	14	f_5, f_3, f_6, f_4, f_{30}, f_{29}, f_{33}, f_{34}, f_{35}, f_{38}, f_{12}, f_{39}, f_{25}, f_{23}
Chi-square	14	f_5, f_3, f_6, f_4, f_{29}, f_{30}, f_{33}, f_{34}, f_{35}, f_{12}, f_{23}, f_{38}, f_{25}, f_{39}
Relief-F	14	f_3, f_{29}, f_4, f_{32}, f_{38}, f_{33}, f_{39}, f_{12}, f_{36}, f_{23}, f_{26}, f_{34}, f_{40}, f_{31}
Multiple filter feature selection (union)	19	f_5, f_3, f_4, f_6, f_{30}, f_{29}, f_{33}, f_{34}, f_{35}, f_{38}, f_{12}, f_{39}, f_{25}, f_{23}, f_{32}, f_{36}, f_{26}, f_{40}, f_{31}
ACO wrapper	10	f_5, f_3, f_4, f_6, f_{12}, f_{23},f_{26}, f_{30}, f_{29}, f_{39}

proposed methodology gave the highest accuracy of 99.86% with a FAR of 0.002. The comparative performance results are shown in Table 4.

To evaluate the effectiveness of ensemble learning techniques, a comparison with the boosting technique was also done. Table 5 depicts the results of the same. It was observed that the bagging classifier performed superlative in comparison to the boosting classifier. Figure 7 depicts the accuracy comparison for both multiple filter with wrapper and multiple filter without wrapper feature selection with bagging and boosting.

The primary objective of this research was to comprehend and characterize the IoT-facilitated real-time abuse and not improving over state-of-the-art (SOTA) anomaly detection techniques. But to better understand how ML-based predictive analytics helps to proactively detect and analyze threats, we compared the results to the recent

Table 4 Performance results

Methodology	Data size	ACC	DR	FAR
MF_EW_Bagging	5K	99.88	99.74	0.0032
	10K	99.84	99.60	0.0025
	15K	99.87	99.97	0.001
MF_E_Bagging	5K	97.99	98.2	2.01
	10K	98.96	98.99	1.06
	15K	98.41	98.82	0.595
IG-ACO_Bagging	Complete	95.3	95.3	0.016
CS-ACO_Bagging	Complete	95.48	95.48	0.021
ReliefF-ACO_Bagging	Complete	96.12	96.0	0.50
IG_Bagging	Complete	94.29	94.24	0.054
CS_Bagging	Complete	95.36	95.42	0.023
ReliefF_Bagging	Complete	96.09	96.0	0.60
ACO_Bagging	Complete	95.38	95.50	0.035

Table 5 Ensemble techniques comparison

Methodology	Data size	ACC	DR	FAR
$MF_EW\_$ Boosting	5K	97.84	98.4	1.89
	10K	97.96	98.5	1.10
	15K	98.41	98.3	0.76
$MF_E_$Boosting	5K	92.90	92.84	6.30
	10K	94.23	94.26	5.22
	15K	94.96	94.36	5.13

SOTA ensemble model [33] which uses Correlation-based feature selection with bat algorithm. The results of the proposed methodology were comparable to the SOTA technique as shown in Fig. 8.

6 Conclusion

With the advancements in technologies over time, the attackers have also come up with novel and potent ways of exploiting our devices and invading our privacy. Using IoT devices as a mode of abuse is an emerging technology challenge which provides new opportunities for abusers to control, harass, and stalk their victims. This paper fostered the need to develop mitigation approaches to prevent, avoid, detect, and predict IoT-based real-time abuse. A set of approaches was put forward, and finally, a prediction model for detecting abnormal use patterns was proffered. The proposed $MF_EW_$Bagging methodology used a multiple filter ensemble with a swarm-based wrapper to reduce the feature set and finally train a bagging classifier. The results were comparable to the existing works with an accuracy of 99.8% on the benchmark NSL-KDD dataset with 10 features selected out of the original (41). Thus, this research recognizes that with the increase in diverse data types and computing power among IoT devices, there is no "one size fits all" cyber-security solution that can protect any IoT deployment though various types of cyber risks. Therefore, with the growing IoT complexity, understanding the risks in a proactive predictive manner is the best way to better defend your networks and systems. As a potential direction of future work, we would like to test other filters and wrappers for feature set reduction. The robustness of the methodology also needs to be evaluated using various available benchmark datasets. As the IoT is resource

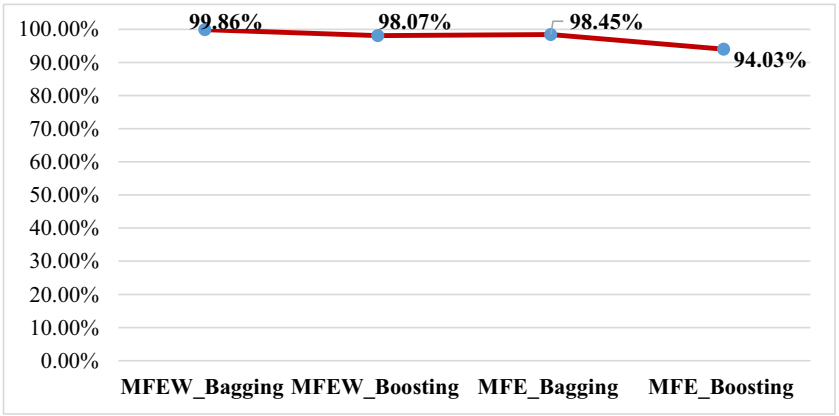

Fig. 7 Accuracy comparison of ensemble classifiers

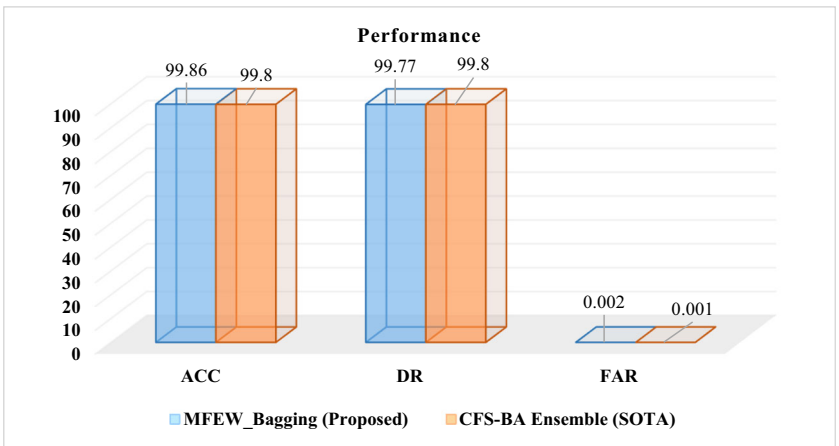

Fig. 8 Performance comparison with SOTA

constrained with power and memory limitations, the energy consumption, processing time, and performance overhead of an IDS are important performance metrics. Thus, robust and lightweight IDS designs for IoT-based smart environments which consider all these factors is the need of the hour.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Blum RW, Nelson-Mmari K (2004) The health of young people in a global context. J Adolesc Health 35(5):402–418
2. Mercy, J. A., Hillis, S. D., Butchart, A., Bellis, M. A., Ward, C. L., Fang, X., & Rosenberg, M. L. (2017). Interpersonal violence: global impact and paths to prevention.
3. Pramod M, Bhaskar CVU, Shikha K (2018) IOT wearable device for the safety and security of women and girl child. Int J Mech Eng Technol 9(1):83–88
4. Ye L, Ferdinando H, Seppänen T, Alasaarela E (2014) Physical violence detection for preventing school bullying. Adv Artif Intell 2014(2014):1–9
5. Vartak GG (2020) Smart security system for women and children using IoT. Adv Innov Res 131
6. Kumar A, Garg G (2019) Sentiment analysis of multimodal twitter data. Multimed Tools Appl 78(17):24103–24119
7. Kumar A, Sachdeva N (2021) Multimodal cyberbullying detection using capsule network with dynamic routing and deep convolutional neural network. Multimedia Systems. https://doi.org/10.1007/s00530-020-00747-5
8. Kumar A, Sachdeva N (2020) Multi-input integrative learning using deep neural networks and transfer learning for cyberbullying detection in real-time code-mix data. Multimedia Systems:1–15
9. Khan S, Gani A, Wahab AWA, Shiraz M, Ahmad I (2016) Network forensics: review, taxonomy, and open challenges. J Netw Comput Appl 66:214–235
10. Visoottiviseth V, Akarasiriwong P, Chaiyasart S, & Chotivatunyu S (2017) PENTOS: penetration testing tool for Internet of thing devices. In TENCON 2017-2017 IEEE Region 10 Conference (pp. 2279-2284). IEEE.
11. Chaabouni N, Mosbah M, Zemmari A, Sauvignac C, Faruki P (2019) Network intrusion detection for IoT security based on learning techniques. IEEE Commun Surv Tut 21(3):2671–2701
12. Buczak AL, Guven E (2015) A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun Surv Tut 18(2):1153–1176
13. Sajith PJ, Nagarajan G (2021) Optimized Intrusion Detection System using computational intelligent algorithm, In: Advances in electronics, communication and computing (pp. 633-639). Springer, Singapore
14. Gendreau AA, Moorman M (2016) Survey of intrusion detection systems towards an end to end secure Internet of things. In: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, Vienna, pp 84–90
15. Sangwan SR, Bhatia MPS (2020) Denigration bullying resolution using wolf search optimized online reputation rumour detection. Procedia Comput Sci 173:305–314
16. Jain DK, Kumar A, Sangwan SR, Nguyen GN, Tiwari P (2019) A particle swarm optimized learning model of fault classification in Web-Apps. IEEE Access 7:18480–18489
17. Ebrahimpour MK, Eftekhari M (2017) Ensemble of feature selection methods: a hesitant fuzzy sets approach. Appl Soft Comput 50:300–312
18. Kumar A, Jaiswal A (2019) Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on twitter. Multimed Tools Appl 78(20):29529–29553
19. Kumar A, Jaiswal A (2020) A deep swarm-optimized model for leveraging industrial data analytics in cognitive manufacturing. IEEE Trans Ind Informat 17(4):2938–2946
20. Yang Y, & Pedersen JO (1997). A comparative study on feature selection in text categorization. In Icml (Vol. 97, No. 412-420, p. 35).
21. Omar N, Jusoh F, Ibrahim R, Othman MS (2013) Review of feature selection for solving classification problems. J Inf Syst Res Innov 3:64–70
22. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. In European conference on machine learning (pp. 171-182). Springer, Berlin, Heidelberg.
23. Sangwan SR, Bhatia MPS (2020) D-BullyRumbler: a safety rumble strip to resolve online denigration bullying using a hybrid filter-wrapper approach. Multimedia Systems:1–17
24. Dorigo M (1992) Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano. Kononenko, I. (1994, April).
25. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications (pp. 1-6). IEEE.
26. Ahmim A, Derdour M, Ferrag MA (2018) An intrusion detection system based on combining probability predictions of a tree of classifiers. Int J Commun Syst 31(9):e3547
27. Xiaofeng Z, & Xiaohong H (2017) Research on intrusion detection based on improved combination of K-means and multi-level SVM. In: 2017 IEEE 17th international conference on communication technology (ICCT) (pp. 2042-2045). IEEE.
28. Omar S, Ngadi A, Jebur HH (2013) Machine learning techniques for anomaly detection: an overview. Int J Comput Appl 79(2):33–41
29. Chkirbene Z, Erbad A, Hamila R, Mohamed A, Guizani M, Hamdi M (2020) TIDCS: a dynamic intrusion detection and classification system based feature selection. IEEE Access 8:95864–95877
30. Xiao Y, Xing C, Zhang T, Zhao Z (2019) An intrusion detection model based on feature reduction and convolutional neural networks. IEEE Access 7:42210–42219
31. Kasongo SM, Sun Y (2019) A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access 7:38597–38607
32. Osanaiye O, Cai H, Choo KKR, Dehghantanha A, Xu Z, Dlodlo M (2016) Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP J Wirel Commun Netw 2016(1):1–10
33. Zhou Y, Cheng G, Jiang S, Dai M (2020) Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput Netw 174:107247
34. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140