International Journal of Safety and Security Engineering
Vol. 15, No. 3, March, 2025, pp. 405-413
Journal homepage: http://iieta.org/journals/ijsse

Optimal Feature Extraction Model for Detecting Cyberattacks on IoT Devices

Monir Abdullah

Computer Science and Artificial Intelligence Department, College of Computing and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia

Corresponding Author Email: mkaid@ub.edu.sa

https://doi.org/10.18280/ijsse.150301

Received: 12 November 2024
Revised: 10 February 2025
Accepted: 18 February 2025
Available online: 31 March 2025

Keywords:
N-BaIoT dataset, IoT, machine learning, cyberattacks, spider monkey optimization (SMO)

ABSTRACT

Internet of Things (IoT) is the technology of this modern era that focuses on connecting devices and sensors to the Internet and can converse with each other without human interaction. IoT technology has been used in many applications, such as smart and wearable devices automatically organizing people's appointments, and in many fields, such as communications in factories and companies. However, in contrast to its rapid spread, it faces several challenges. In terms of privacy, security, and confidentiality, IoT is vulnerable to many types of cyberattacks, making it necessary to develop safe solutions to secure IoT networks. In this paper, two feature extraction algorithms are integrated with three machine learning (ML) techniques to develop a model that detects a cyberattack faced by IoT devices. Butterfly optimization algorithm (BOA) and spider monkey optimization (SMO) with Naïve Bayes (NB), Random Forest (RF), and K-Nearest Neighbor (KNN) models are implemented. The experiment was conducted on the N-BaIoT dataset containing more than 800,000 records covering 10 IoT device attacks. The results show that the SMO feature extraction algorithm with the KNN classifier model outperformed other algorithms and achieved 100% in all performance metrics.

1. INTRODUCTION

IoT is classified as a technology that affects people positively. In daily activities in various areas of life, it has improved the quality of life and provides the best methods for communicating with various kinds of devices via the Internet. For example, in the healthcare sector, the data are collected by sensors and devices measuring the heartbeat and blood pressure levels. However, IoT attacks may cause privacy violations and threaten people's lives and privacy [1]. The healthcare sector has faced many electronic attacks. The existing data are sensitive data, and their exposure to the attack endangers patients' lives, so the privacy and security of the information must be appropriately applied. The IoT field has received great attention, and IoT is growing on a large scale. IoT systems include a wide range of technologies and will continue to in the coming years. However, there are still many new and severe challenges to the problem of security and privacy in this field. In this research, a model using the ML algorithm will be built to discover cyberattacks targeting IoT devices and networks by determining the type of attack [1]. IoT security is different from other forms of computing in many ways, especially in the challenges of other computing devices like desktops, laptops, servers, or mobile devices. For all IoT systems, the authentication mechanism is crucial for security and privacy [2]. The N-BaIoT dataset attack types are summarized into the following [3]:

DoS/DDoS: A type of attack in which an attacker causes a computing or memory resource to be overloaded to the extent that no authentic request gets through due to congestion.

Man-in-the-Middle Attack: An intruder intercepts the messages between two devices so that a potential intruder may have eavesdropping capability along with data modification during transit. This could be made possible by several means, including address resolution protocol (ARP) spoofing and domain name systems (DNS) poisoning. Thus, hackers steal sensitive information and modify commands to effect unauthorized actions.

Backdoor Attack: A backdoor software installation vulnerability in a device is exploited by an attacker for constant and unauthorized access. This can allow the backdoor to continuously monitor or control the device without being noticed. It can be lessened through periodic security audits and assurance of software integrity through checksums.

Probing: This includes probing attacks an intruder uses to scan the network for information gathering or determining possible vulnerabilities. A mapped sketch of available machines and services on the network provides data for future exploits.

Apart from these, replay attacks, password cracking, and injection attacks would emulate other types of security threats an IoT device would generally face.

These are the main cyberattack detection methods:

(1) Modern technologies contribute to the protection and security of information by monitoring and analyzing cyberattacks.

(2) That requires updating and developing antivirus programs, determining the most sensitive points prone to

penetration, and directly confronting cyberattacks.

(3) The measures necessary to restrict the attacks on cyberspace include increasing the level of awareness among citizens, infrastructure development, and keeping pace with techno-development.

(4) The imperative of developing a rapid response plan and dealing with cybersecurity incidents [4].

In our research, an ML-based model to detect cyberattacks targeting IoT devices and networks is proposed. Moreover, the proposed model identifies the type of attacks by analyzing more than 800,000 contact data related to various IoT device attacks. The main contributions of this article are as follows:

(1) Adapting butterfly optimization algorithm (BOA) and spider monkey optimization (SMO) for dataset feature extraction.

(2) Integrating BOA and SMO with ML classifier to detect cyberattacks in IoT devices.

(3) Evaluating the proposed model by applying several ML performance metrics.

The remainder of this paper is divided into the following sections. Related works of different ML classification models, feature selection algorithms, ML-based cyberattack detection models, and the N-BaIoT cyberattack dataset are presented in Section II. Section III presents the proposed model implementation in detail. The results are discussed in Section IV, and Section V concludes the article.

2. RELATED WORKS

The IoTs is the new discipline that not only takes care of scientific, engineering and technical aspects but also integrates social sciences and analyzes big data derived from social media. Recently, detection systems have become the prime focus of researchers in IoT environments due to the ever-increasing threat of botnet attacks on such devices. It effectively addresses network intrusion detection systems (IDSs) for protection against malicious activities on networks. Most public detection systems operate based on attack signatures; they are called signature-based detection systems. These systems recognize all the known types of attacks by matching the pattern of incoming attacks with pre-recorded signatures. They require a robust infrastructure and sophisticated tools since many signature rules must be added to their databases. Various research works have proven that ML techniques can be of great assistance in attack detection tasks. It is already known that most attacks in the IoT environment are of a botnet nature, and several IoT devices still present vulnerabilities owing to limited memory and computational resources, which seem to be an obstruction to strong security mechanisms. Moreover, attackers can bypass many rule-based detection systems easily. Researchers have explored the development of an ML-based attack detection system featuring a sequential detection architecture. They employed an efficient feature selection approach to develop a lightweight, high-performance detection system that achieved improved overall detection results [5]. They also proposed a framework for specifying an effective algorithm to detect malicious activity in IoT using ML, where a Naïve Bayes (NB) model was observed to perform well in anomaly detection. A cyberattack detection system targeting sewage IoT devices was also proposed, which managed to achieve an accuracy of 92% in fixed scenarios and 72% in mobile environments [6]. In reference [7], results have been produced by testing and validating four binary classifiers: decision trees (DT), extra trees classifiers, Random Forest (RF), and support vector machines (SVMs). The RF classifier outperformed all other classifiers when it was trained on a specific device and used to test the anomalies that come from completely unrelated devices.

2.1 Feature extraction models

2.1.1 Butterfly optimization algorithm (BOA) model

BOA is a novel optimization technique inspired by how butterflies forage for and attract mates. In this algorithm, the behavior of females using their chemoreceptors, an external sensory organ present in many parts of the body, is modeled. These chemoreceptors help in perceiving flower or food odors, as well as in identifying the best possible mating partners. When the butterflies are in movement, they diffuse the odor in variated concentrations, and this scent directs the movement of the search agents (the butterflies) in the BOA algorithm. For instance, where a butterfly cannot scent others' fragrances within a search area, it will tumble and reposition randomly. However, when a butterfly stabilizes and only detects the perfume from the most successful butterfly it knows, it will try to move toward that butterfly [8]. The BOA has shown promise in feature selection tasks across various studies. For instance, in reference [9], the monarch BOA algorithm is applied to feature selection, achieving high classification accuracy (93%) while significantly reducing feature set size. Moreover, binary variants of BOA demonstrate improved classification accuracy compared to other wrapper-based algorithms [10].

2.1.2 Spider monkey optimization (SMO) model

The SMO algorithm draws inspiration from the social and foraging behaviors of spider monkeys. It uses a fission-fusion social structure where the monkeys form and dissipate groups of different sizes. Some primary characteristics that define spider monkeys in the SMO algorithm include the following [11]:

(1) Each group contains 40 to 50 monkeys; these are called individuals within the framework of SMO.

(2) Among the troop of monkeys, there is a global leader (GL) who can divide them into subgroups of 3-8 when food is insufficient, where each subgroup will then forage independently.

(3) Each subgroup is usually led by a local leader (LL), who usually guides foraging.

(4) Members of the group use particular sounds to communicate with one another and promote social behavior.

For the SMO algorithms, a hybrid approach combining SMO with simulated annealing and ReliefF filtering demonstrated superior performance in identifying biomarker genes from cancer datasets, achieving up to 99.45% accuracy [12]. In fuzzy classifier construction, binary SMO was employed for feature selection, while continuous SMO optimized fuzzy rule antecedents, resulting in classifiers with minimal rules and reduced features while maintaining competitive accuracy [13]. These studies highlight the effectiveness of SMO-based approaches in feature selection across diverse applications. The SMO algorithm is presented in Figure 1.

N-BaloT Dataset Optimization Algorithm

1. **Initialize Dataset:**
   - Randomly initialize the swarm of N monkeys.
   - Set the Perturbation rate and limit for LL and GL.
2. **Measure Fitness:**
   - Measure the fitness of the individuals.
3. **Select LLs and GLs**:
   - Use a greedy selection process to elect LLs and GLs.
4. **Stopping Criteria**:
   - While stopping criteria are not met:
     1. Obtain new positions for all individuals using the LL phase.
     2. Obtain the fitness values of each group member to drive the greedy selection.
     3. Update the locations of GLs and LLs based on fitness.
   - If there is no change in any L group leader for a predefined limit:
     - Apply the LL Decision (LLD) phase.
     - If GL is not predefined, apply the GL Decision (GLD) phase.
     4. Maintain the minimum size of each group.
5. **Return Best Solution:**
   - Return the best solution found by the SMO algorithm.

Figure 1. Spider monkey optimization (SMO) algorithm

Table 1. SMO-related works

Ref.	Feature Selection by SMO Model	Year	ML Classifier	Dataset	Accuracy (%)
[13]	Binary SMO (BSMO)	2019	Fuzzy Classifier	38 Datasets	99.45
[14]	Oscillating SMO	2021	SVM, LDA, KNN, and RF	Soil image dataset	82.25
[15]	Conditional Random Field (CRF) & SMO	2021	CNN	NSL-KDD Dataset	99
[16]	Gaussian Mutation-SMO(GM-SMO)	2022	ANN, CNN	AID, UCM, NWPU45 dataset	Highest 99.46
[17]	Self-Improved Standard SMO	2022	Deep Learning Belief Network	COVID-19 Image Dataset	90.5
[18]	SMO	2023	CNN-LSTM	Dementia sufferers images	89.72
[19]	Optimal feature extraction is achieved using a differential SMO	2023	NB, RF, SVM, DT	Magnetic resonance imaging (MRI) datasets	NB = 91, RF = 94 SVM = 96 DT = 93.5
[20]	Cuckoo search algorithm (CSA) & SMO	2024	ReliefF+ PCA	8 Cancer datasets	90.6

The SMO algorithm has many applications for feature selection over different datasets concerning accuracy and classification performance. In 2019, a fuzzy classifier developed with binary SMO reached an accuracy of 99.45% for 38 datasets, showing the efficiency of SMO in optimizing fuzzy classifiers [13]. It obtained an accuracy of 82.25% with a dataset of soil images taken in 2021 by employing oscillating SMO fine-tuning for feature selection in an ensemble of classifiers comprised of SVMs, linear discriminant analysis, K-Nearest Neighbor (KNN), and RF [14]. The accuracy of intrusion detection reached 99% using the proposed CNN model combined with conditional random field and SMO on the NSL-KDD dataset that enhances IoT security [15]. The highest value recorded was 99.46% in 2022 when both ANN and CNN models adopted GM-SMO in recognizing remote sensing scenes [16]. The adaptability of SMO flowed to medical image analysis, with the accuracy of a deep belief network with self-optimizing SMO reaching 90.5% in COVID-19 prediction, while that of a CNN-LSTM framework that employed SMO reached an accuracy of 89.72% in dementia detection [17, 18]. Furthermore, in MRI datasets, differential SMO improved feature extraction such that classification accuracy for detecting a particular lumbar spine disease reached 96% by SVM in 2023 [19]. Lastly, in 2024, the Cuckoo Search algorithm combined with SMO was proposed for the detection of cancer and reported 90.6% accuracy on eight different cancer datasets [20]. These results themselves indicate that SMO has always been able to enhance the performance of ML in a continuous manner, from cybersecurity to healthcare. SMO-related works are presented in Table 1.

2.2 ML-based models

2.2.1 Naïve Bayes classifier

In statistics, NB classifiers are a family of simple "probabilistic classifiers" based on the application of Bayes' theorem with (Naïve) assumptions about independence between features. These are some of the simplest Bayesian network models but can be combined with kernel density estimation to achieve higher levels of accuracy. The NB classifier scales exceedingly well. There are only a few parameters to be estimated, which scales linearly with the number of variables (features/predictors) in the learning problem. Maximum likelihood training can be done very efficiently by evaluating a closed-form expression that takes linear time rather than an expensive iterative approximation, as used by many other classifiers. It is also referred to in the literature of statistics and computer science as NB models and independent Bayes [21]. Recent research has addressed the use of ML techniques, especially the NB model, for network attack detection. A Gaussian NB model was explored in classifying cyberattacks in streaming data, focusing on its adaptability [22]. The hybrid approach proposed in reference [23] combines the heuristics clustering algorithm and NB model to detect DDoS attacks, which showed improved accuracy and detection rates. In addition, three models, including NB, RF, and stochastic gradient boosting, have been compared for

DDoS attack classification. Among them, stochastic gradient boosting proved to be the most accurate at 100% [24]. Recently, a probability-based supervised ML algorithm has been introduced NB model for intrusion detection on the UNSW-NB15 dataset [25]. All these studies indicate the potency of the NB model and other ML techniques in detecting various network attacks; hence, research is ongoing to further improve their accuracy and adaptability against evolving cyber threats.

2.2.2 Random Forest (RF) classifier

RF is flexible, easy to use, and gives the best results most of the time without hyperparameter tuning. Because of its simplicity and efficiency, it is among the most used algorithms since it can be applied to ranking and regression. RF is one of the algorithms for supervised learning. "Jungle" is an ensemble of DTs usually trained by the "bagging" method. The general intuition of this approach is that the ensemble of learning models boosts the final outcome. In other words, RFs construct a large set of DTs and combine their predictions to produce more accurate and robust estimates. Among many advantages, one of the most important benefits of RF is that it can be used for both classification and regression problems, which constitute the biggest part of all ML systems today [26]. Recent works have discussed the adoption of RF for detecting network attacks, as in reference [27]. RF is combined with principal component analysis to enhance attack detection for IoT devices, reaching as high as 99.2% accuracy [28]. In previous study [29], an RF-based model was developed for DDoS attack detection. The Gini Index and entropy criteria are used in the model to improve its accuracy. For instance, ET-RF [30] is proposed, attaining an accuracy of 99% in the CICDDoS2019 dataset. In previous study [31], RF is utilized on the NSL-KDD dataset for intrusion detection. As such, feature selection should be done with the Gini importance that improves the effectiveness of the model.

2.2.3 K-Nearest Neighbor (KNN) classifier

The KNN algorithm is the simplest form of ML based on the technique of supervised learning. KNN assumes that new cases or data points would be similar to the existing ones and assigns them to the most similar categories. In this algorithm, it retains all the incoming data and classifies any new data point by comparing its similarity with the stored examples. This will easily classify any incoming data into the right range of classes. Though KNN finds its applications in both regression and classification, the most common area of application is classification. KNN is a nonparametric algorithm that does not make any assumptions on the underlying distribution of data. It is often referred to as a lazy learning algorithm because it only stores the dataset in memory and does the classification when needed, rather than learning in advance from the training set. During the training phase, KNN only stores the dataset, and when new data shows up, it classifies based on similarity to existing data points [32]. Recently, several works investigated the performance of KNN for attack detection. KNN tends to have high accuracy and performance metrics while detecting different cyberattacks. Previous study [33] reported 99.996% in binary classification and 99.988% in multi-class classification with the NSL-KDD dataset. In this direction, it has been seen that KNN performs best in the CICIDS 2017 dataset, among other supervised learning algorithms, for the highest F1-score and accuracy [34]. In reference [35], an anomaly-based detection model developed using KNN achieved 92% accuracy, 100% precision, and 95.8% F1-score in detecting denial-of-service attacks. Although KNN performs well when it comes to the capability of detection, it may be inefficient when it comes to bits' time complexity. Overall, these studies support KNN as a feasible and efficient algorithm for intrusion detection in a network.

2.3 N-BaIoT cyberattacks dataset

The N-BaIoT dataset was utilized to validate the proposed IDSs. The ensemble averaging deep neural networks achieved the target of attack detection by botnets in heterogeneous IoT devices with an average accuracy of 97.21%, precision of 91.41%, recall of 87.31%, and an F1-score of 88.48% [36]. AdaBoost and eXtreme gradient boosting (XGBoost) models have been implemented to meet particular security challenges in IoT networks [37]. The N-BaIoT dataset, which was produced by the injection of the Bashlite and Mirai botnet attacks on various IoT devices, has widely been used in research on the detection of IoT botnets [38, 39]. This dataset addresses the scarcity of publicly available botnet-specific datasets in IoT domains [40]. Various research studies have designed and built effective detection models based on ML and DL methodologies, of which most reported that RF and gradient boosting reliably showed higher accuracy. Feature selection methods, such as the Fisher Score or PCA, were considered to optimize the effectiveness of the detection performance [40]. It was confirmed that the N-BaIoT dataset indeed outperformed the existing wired datasets, such as NSL-KDD, because it covers IoT-specific attacks and considers relevant network layers [41]. Research supports the idea that not all the features are necessary for effective detection, which could reduce the detection time for ML models.

In our research, BOA and SMO feature selection algorithms with ML-based models are integrated to detect cyberattacks targeting IoT devices. Moreover, the proposed model identifies the type of attacks by analyzing more than 800,000 contact data related to various IoT devices and attacks.

3. PROPOSED MODEL IMPLEMENTATION

The experimental study in this work was conducted on a desktop running Microsoft Windows 10 Home (64-bit). The machine used for the implementation is Alienware Aurora R9, with an octa-core Intel® Core™ i9-9900 processor that represents a base frequency of 3.10 GHz. It has 32 GB RAM and a 1 TB hard drive. The software requirements are fulfilled using Keras and Python programming tools. Detailed hardware and software specifications for the proposed integrated model are illustrated in Table 2.

Table 2. System configuration for the proposed model implementation

Processor	Core i9
GPU	NVIDIA 4 GB
OS	Windows 10 - 64 Bit
RAM	32GB
Language	Python
Software	Numpy, TensorFlow, Scikitlearn, Pandas

3.1 Experiment framework

The experiment framework was created to obtain a clear perspective of our experiment design on the problem area as described in Figure 2.

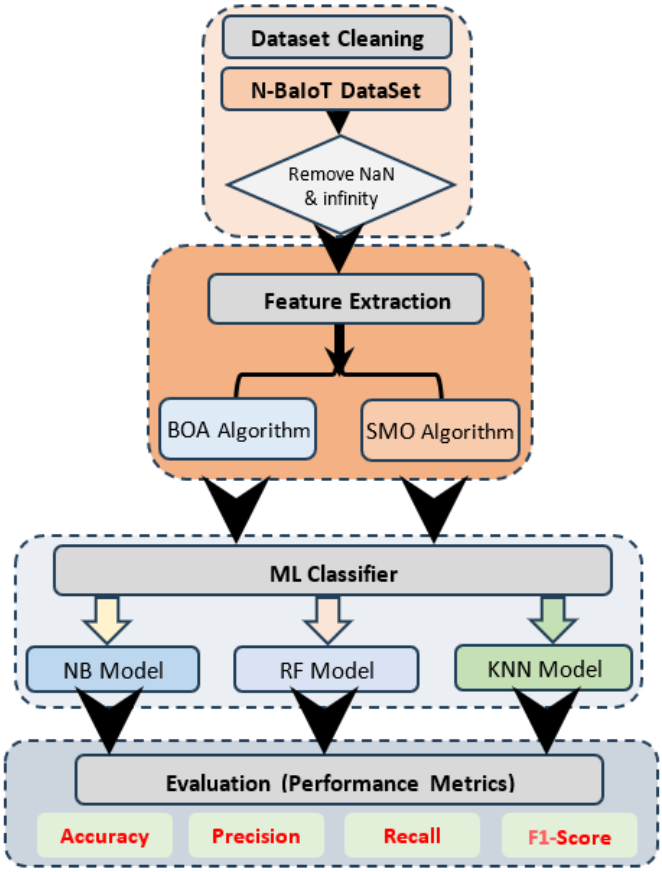

Figure 2. The proposed framework

The framework consists of four phases: dataset cleaning, feature extraction, ML classifiers, and evaluation.

The flowchart illustrates the process of building and evaluating an ML model on the N-BaIoT dataset. Below is a step-by-step description of each stage:

Dataset Cleaning (N-BaIoT Dataset): The input is the N-BaIoT dataset, and the data is cleaned, where missing or NaN values are removed.

Feature Extraction: The dataset is first cleaned; then, the features are extracted for further processing. The techniques used are BOA and SMO algorithms.

ML Classifiers: These extracted features will be fed to several ML classifiers. NB, RF, and KNN models shall be implemented.

Model Evaluation: The models will be evaluated in terms of precision, accuracy, recall, and F1-score.

3.2 Dataset preparation

N-BaIoT focuses on the network-based detection of IoT botnet attacks. Additionally, it closes the gap with respect to publicly available botnet datasets related to IoT environments. Earlier works regarding the detection of IoT botnets or IoT traffic anomalies relied on emulated or simulated data. In contrast, this dataset allows empirical evaluations using real traffic data captured from nine commercial IoT devices infected with genuine botnets of two different families in a contained network.

3.3 Feature extraction

Feature selection is probably one of the most important concepts in ML, as it has a high influence on the performance of our model. The functional attributes of data utilized for training the ML models are core contributors in terms of the performance achievable from them. The features that are irrelevant or only partially relevant might have adverse implications on model effectiveness. Therefore, feature selection and data cleaning should be the initial and primary steps in model design. BOA and SMO are two feature extraction algorithms that we will implement in our research, shown in the above framework. To design an SMO fitness function for feature selection on a given dataset, the goal would be to identify an optimal subset of features that balances high classification accuracy with a minimum number of selected features. In general, both these objectives are included in the fitness function. A typical fitness function (ft) for feature selection in SMO can be represented as Eq. (1):

$$ft(sf) = \alpha \cdot (1 - \text{Accuracy}(sf)) + \beta \cdot (|sf| / |f|) \quad (1)$$

where, *sf* is the subset of selected features, |sf| is the number of features in subset *sf*, |*f*| is the total number of features in the dataset, Accuracy(*sf*) is the classification accuracy of the model trained using subset *sf*, and α and β are the weight parameters that control the importance of classification accuracy and feature subset size.

3.4 Model training and testing

The purpose of training the model is to overcome problems such as overfitting when the model remembers the dataset pattern. Making a bad choice in splitting the dataset can lead to unwanted results. Therefore, by separating the training data from the test data for evaluation, we can test our model with data samples that were not previously used. In our research, the data are split into a 70/30 ratio. That is, 70% of the data are used to train the model, and the remaining 30% are used to test the model. Three ML classifier models are used: NB, RF, and KNN.

3.5 Performance metrics

This will be a necessary step in verifying the learning of our model and determining its performance. The model's evaluation would include four different performance metrics: accuracy, precision, F1-score, and recall. Performance metrics using the formulas below were calculated based on the confusion matrix below:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (2)$$

where, *TP* represents the number of true positives, *TN* is the number of true negatives, *FP* refers to the number of false positives, and *FN* refers to the number of false negatives.

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

$$F1\_Score = 2 * \frac{(Precision * Recall)}{(Precision * Recall)} \quad (5)$$

4. RESULTS AND DISCUSSION

Our research presented an ML-based model to detect cyberattacks targeting IoT devices and networks. The proposed model is classified and evaluated based on their performance, where all available features are used in this classification based on the best-featured multiclass classification task where only the most relevant features are used. Classification analysis is completed by the classifier to discuss the remarkable behavior. Moreover, the proposed model identifies the type of attacks by analyzing more than 800,000 contact data related to various IoT devices and attacks.

In Table 3, performance measures obtained from three ML classifiers, NB, RF, and KNN, are detailed. These three algorithms have run their vanilla model and have been optimized by BOA and SMO algorithms for feature selection. The metrics used in assessing the models are accuracy, precision, recall, and F1-score, where the effectiveness of each classifier before and after optimization can be estimated.

All the performance metrics for NB, RF, and KNN classifiers improved significantly with various optimization techniques applied. KNN always has a higher score than other models, with its non-optimized version scoring very high (Accuracy = 0.982, F1-score = 0.969) while reaching perfection after SMO optimization. The RF classifier improved from the previous non-optimized settings to 0.942 accuracy, with a respectable F1-score of 0.935. While NB indicates the poorest initial performance, it also improved with optimization, and SMO models boosted its F1-score from 0.789 to 0.880. In all cases, SMO outperforms BOA in terms of accuracy and F1-score; therefore, SMO can be said to be a better optimization method for these classifiers. Figure 3 presents the confusion matrix for the KNN model.

The matrix shows how well the model differentiates between benign traffic and various types of botnet attacks, such as gafgyt and mirai, where most of the predictions correctly lie along the diagonal, which essentially reflects correct classification. Misclassifications are minimal according to the sparsity of the off-diagonal elements. As the confusion matrix from the KNN model indicates, the perfect accuracy of prediction across all classes is as follows.

Table 3. Performance metrics for the three ML classifiers with SMO

ML Classifiers	Feature Selection Models	Accuracy	Precision	Recall	F1-Score
NB Classifier	Non-Optimized	0.847	0.775	0.833	0.789
	BOA Models	0.857	0.797	0.857	0.817
	SMO Models	0.897	0.887	0.907	0.880
RF Classifier	Non-Optimized	0.871	0.919	0.878	0.841
	BOA Models	0.893	0.928	0.893	0.855
	SMO Models	0.942	0.932	0.963	0.935
KNN Classifier	Non-Optimized	0.982	0.979	0.968	0.969
	BOA Models	0.995	0.994	0.987	0.986
	SMO Models	1.0	1.0	1.0	1.0

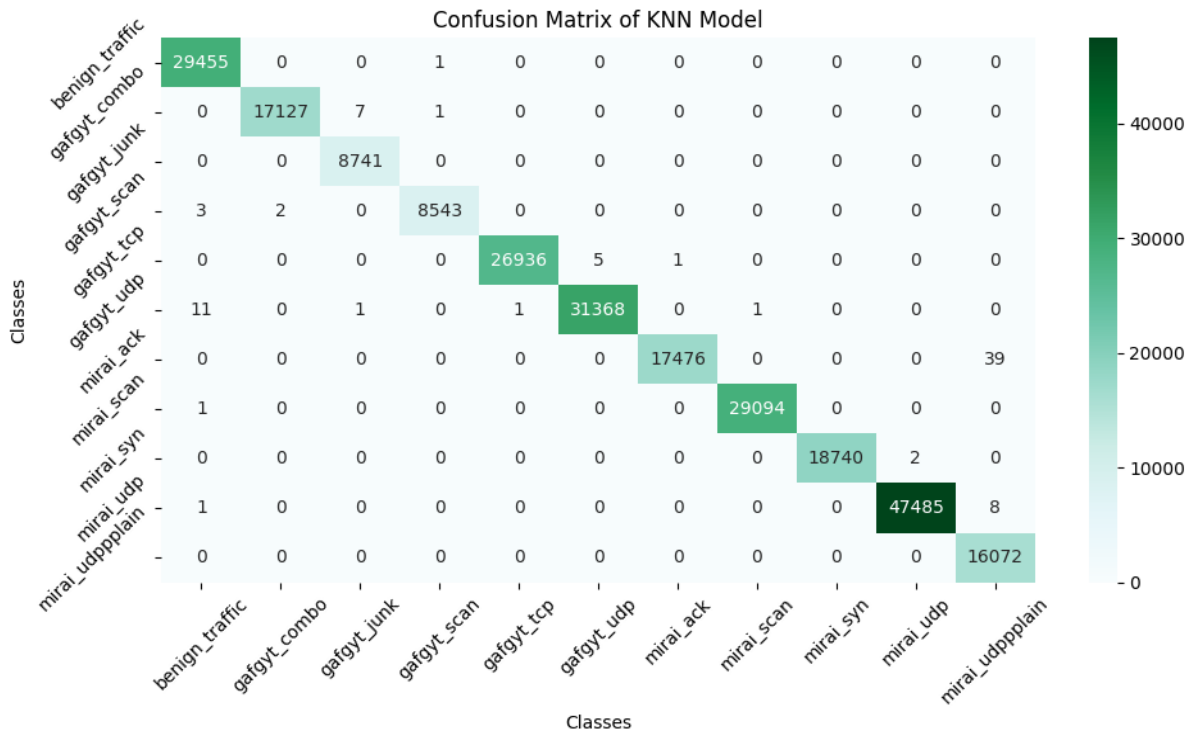

Figure 3. Confusion matrix of the KNN model with SMO

This includes 29,455 observations for benign_traffic, all of which are correctly predicted, and 17,127 for gafgyt_combo, also fully predicted. There is a correct prediction of 8,741 for gafgyt_junk, 8,543 for gafgyt_scan, 26,936 for gafgyt_tcp, and 31,368 for gafgyt_udp. Among the mirai classes, the model predicts 17,476 for mirai_ack, 29,094 for mirai_scan, 18,740 for mirai_syn, 47,485 for mirai_udp, and 16,072 for mirai_udpplain, where every class has its observation correctly classified.

The results show that the KNN + SMO classifier excelled in all the metrics, with perfect scores; this could suggest that it captured the pattern in the dataset very well. However, this could raise eyebrows regarding overfitting. The RF + SMO classifier is in good standing in performance, where precision and recall are high, showing that the balancing between identifying positive cases and minimizing false alarms worked well. Whereas the NB + SMO classifier presents a good performance, it also somewhat lags behind, with lower accuracy and F1-score, which is indicative of fields that may harbor improvement as per Table 4 and Figure 4.

The performance of different ML models on the N-BaIoT dataset is presented for the detection of cyberattacks. This discussion covers critical metrics, including accuracy, precision, recall, and F1-score, and thus involves insight into each model's effectiveness. In particular, perfection scores across all metrics were achieved for the integrated proposed method in 2024; this, therefore, shows the progress being made within ML techniques in cybersecurity, as defined in Table 5.

The proposed integrated SMO-KNN model has perfect performance, with 100% accuracy, precision, recall, and F1-score, which indicates faultless classification. Whereas in comparison, deep neural network-long short-term memory performs nearly perfectly, though slightly lower in precision and recall [41]. KNN, RF, and NB have good accuracy but perform poorly in terms of precision [41]. The improved Harris Hawks optimization algorithm for neural networks (IHHO-NN) has balanced but lower scores [42]. In reference [39], XGBoost delivers consistent performance with 99% across all metrics, and the IHHO-NN also shows robust results, though slightly lower than the SMO-KNN model [39]. Overall, SMO-KNN stands out as the top-performing model with optimal performance. This means that the algorithm increases its accuracy in finding the best features and speeds up its convergence to the optimal solution [14]. It enhances SMO to be more robust and reliable, especially when the complication of optimization settings makes conventional methods inefficient.

Table 4. Performance metrics of various integrated models

Integrated Model	Accuracy	Precision	Recall	F1-Score
NB + SMO	0.897	0.887	0.907	0.880
RF + SMO	0.942	0.932	0.963	0.935
KNN + SMO	1.0	1.0	1.0	1.0

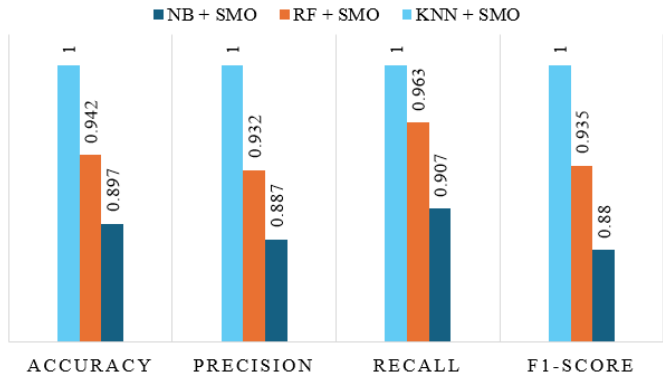

Figure 4. Performance of integrated ML classifier with SMO model

Table 5. Performance metrics of various ML models on N-BaIoT dataset

Ref.	Year	Algorithm(s)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[41]	2019	DNN-LSTM	99.96	99.77	99.66	99.66
[42]	2021	KNN, RF, NB	99.00	86.65	99.00	99.00
[43]	2023	IHHO-NN	98.07	97.04	98.73	97.87
[44]	2023	HMMLB-BND	99.43	99.13	99.12	99.13
[39]	2024	XGBoost	99.00	99.00	99.00	99.00
Proposed Model	2024	Integrated (SMO-KNN)	100	100	100	100

5. CONCLUSION AND FUTURE WORKS

In this research, three ML classifiers were implemented to classify cyber security attacks against IoT devices. The three classifier models were successfully integrated with two feature selection algorithms and produced optimal results. The experiment study has been conducted for KNN, NB, and RF classifier models on an N-BaIOT dataset. The experimental results showed that the KNN with SMO feature selection algorithm performed better than other models with an accuracy as high as 100%. Additionally, our proposed integrated model is compared with other robust and state-of-the-art detection schemes. In the future, we can implement the integrated models in different related problems and datasets.

ACKNOWLEDGEMENT

The authors are thankful to the Deanship of Graduate Studies and Scientific Research at the University of Bisha for supporting this work through the Fast-Track Research Support Program.

REFERENCES

[1] Chen, K., Zhang, S., Li, Z., Zhang, Y., Deng, Q., Ray, S., Jin, Y. (2018). Internet-of-Things security and vulnerabilities: Taxonomy, challenges, and practice. Journal of Hardware and Systems Security, 2: 97-110. https://doi.org/10.1007/s41635-017-0029-7

[2] Fatayer, T.S., Azara, M.N. (2019). IoT secure communication using ANN classification algorithms. In 2019 International Conference on Promising Electronic Technologies (ICPET), Gaza, Palestine, pp. 142-146. https://doi.org/10.1109/ICPET.2019.00033

[3] Sasi, T., Lashkari, A.H., Lu, R., Xiong, P., Iqbal, S. (2024). A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges. Journal of Information and Intelligence, 2(6): 455-513. https://doi.org/10.1016/j.jiixd.2023.12.001

[4] Khan, R.U., Kumar, R., Alazab, M., Zhang, X. (2019). A hybrid technique to detect botnets, based on P2P traffic similarity. In 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, VIC, Australia, pp. 136-142. https://doi.org/10.1109/CCC.2019.00008

[5] Haq, S., Singh, Y. (2018). Botnet detection using machine learning. In 2018 Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC), Solan, India, pp. 240-245. https://doi.org/10.1109/PDGC.2018.8745912

[6] Joshi, S., Abdelfattah, E. (2020). Efficiency of different machine learning algorithms on the multivariate classification of IoT botnet attacks. In 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, pp. 517-521. https://doi.org/10.1109/UEMCON51285.2020.9298095

[7] Arora, S., Singh, S. (2019). Butterfly optimization algorithm: A novel approach for global optimization. Soft Computing, 23: 715-734. https://doi.org/10.1007/s00500-018-3102-4

[8] Alweshah, M., Khalaileh, S.A., Gupta, B.B., Almomani, A., Hammouri, A.I., Al-Betar, M.A. (2022). The monarch butterfly optimization algorithm for solving feature selection problems. Neural Computing and Applications, 34: 11267-11281. https://doi.org/10.1007/s00521-020-05210-0

[9] Arora, S., Anand, P. (2019). Binary butterfly optimization approaches for feature selection. Expert Systems with Applications, 116: 147-160. https://doi.org/10.1016/j.eswa.2018.08.051

[10] Bansal, J.C., Sharma, H., Jadon, S.S., Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. Memetic Computing, 6: 31-47. https://doi.org/10.1007/s12293-013-0128-0

[11] Sahu, B., Panigrahi, A., Dash, B., Sharma, P.K., Pati, A. (2023). A hybrid wrapper spider monkey optimization-simulated annealing model for optimal feature selection. International Journal of Reconfigurable and Embedded Systems, 12(3): 360-375. https://doi.org/10.11591/ijres.v12.i3.pp360-375

[12] Hodashinsky, I.A., Nemirovich-Danchenko, M.M., Samsonov, S.S. (2019). Feature selection for fuzzy classifier using the spider monkey algorithm. Бизнес-Информатика, 13(2): 29-42. https://doi.org/10.17323/1998-0663.2019.2.29.42

[13] Agarwal, R., Shekhawat, N.S., Kumar, S., Nayyar, A., Qureshi, B. (2021). Improved feature selection method for the identification of soil images using oscillating spider monkey optimization. IEEE Access, 9: 167128-167139. https://doi.org/10.1109/ACCESS.2021.3135536

[14] Parimala, G., Kayalvizhi, R. (2021). An effective intrusion detection system for securing IoT using feature selection and deep learning. In 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, pp. 1-4. https://doi.org/10.1109/ICCCI50826.2021.9402562

[15] Shaik, A.L.H.P., Manoharan, M.K., Pani, A.K., Avala, R.R., Chen, C.M. (2022). Gaussian mutation-spider monkey optimization (GM-SMO) model for remote sensing scene classification. Remote Sensing, 14(24): 6279. https://doi.org/10.3390/rs14246279

[16] Rao, J.M., Narayan, B.H. (2022). Novel coronavirus (COVID-19) prediction using deep learning model with improved meta-heuristic optimization approach. In 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 935-943. https://doi.org/10.1109/ICSSIT53264.2022.9716478

[17] Sweety, K., Nagalakshmi, M. (2023). A robust deep neural network framework for the detection of dementia. In 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, pp. 686-691. https://doi.org/10.1109/ICPCSN58827.2023.00119

[18] Singh, D., Singla, J., Rahmani, M.K.I., Ahmad, S., et al. (2023). Lumbar spine disease detection: Enhanced CNN model with improved classification accuracy. IEEE Access, 11: 141889-141901. https://doi.org/10.1109/ACCESS.2023.3342064

[19] Rajasekar, M., Arunachalam, P., Priyadharsini, P., Devi, N.L., Abbas, H.H., Al-Qaisy, S.A. (2024). An optimized framework development of ABC algorithm along with SVMP algorithm for lung cancer detection. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, pp. 184-187. https://doi.org/10.1109/ICACITE60783.2024.10616706

[20] Hasan, M., Islam, M.M., Zarif, M.I.I., Hashem, M.M.A. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things, 7: 100059. https://doi.org/10.1016/j.iot.2019.100059

[21] Desai, P. (2024). Enhancing cybersecurity through Bayesian node profiling and attack classification. International Journal of Wireless and Microwave Technologies, 14(1): 43-51. https://doi.org/10.5815/ijwmt.2024.01.04

[22] Bista, S., Chitrakar, R. (2017). DDoS attack detection using heuristics clustering algorithm and Naïve Bayes classification. Journal of Information Security, 9(1): 33. https://doi.org/10.4236/jis.2018.91004

[23] Firmansyah, R., Utami, E., Pramono, E. (2022). Evaluation of naïve bayes, random forest and stochastic gradient boosting algorithm on DDoS attack detection. Proceeding International Conference on Information Science and Technology Innovation, 1(1): 92-97. https://doi.org/10.35842/icostec.v1i1.16

[24] Sonule, A.R., Kalla, M., Jain, A., Chouhan, D.S. (2021). Detection of network attacks using machine learning: A new approach. International Journal for Research in Applied Science & Engineering Technology, 9(12): 1881-1890. https://doi.org/10.22214/ijraset.2021.39640

[25] Wu, Y., He, X., Chen, X. (2022). IoT-botnet traffic detection based on deep forest. In 2022 IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, pp. 1388-1393.

https://doi.org/10.1109/ICCT56141.2022.10072774
[26] Pirtama, A., Prasetia, Y., Saputra, R.I., Winanto, E.A. (2024). Improvement attack detection on internet of thinks using principal component analysis and random forest. Media Journal of General Computer Science, 1(1): 14-19. https://doi.org/10.62205/mjgcs.v1i1.8
[27] Chu, T.S., Si, W., Simoff, S., Nguyen, Q.V. (2022). A machine learning classification model using random forest for detecting DDoS attacks. In 2022 International Symposium on Networks, Computers and Communications (ISNCC), Shenzhen, China, pp. 1-7. https://doi.org/10.1109/ISNCC55209.2022.9851797
[28] Lahasan, B., Samma, H. (2022). Optimized deep autoencoder model for internet of things intruder detection. IEEE Access, 10: 8434-8448. https://doi.org/10.1109/ACCESS.2022.3144208
[29] Gaur, V., Kumar, R. (2022). ET-RF based model for detection of distributed denial of service attacks. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, pp. 1205-1212. https://doi.org/10.1109/ICSCDS53736.2022.9760938
[30] Negandhi, P., Trivedi, Y., Mangrulkar, R. (2019). Intrusion detection system using random forest on the NSL-KDD dataset. In Emerging Research in Computing, Information, Communication and Applications: ERCICA 2018, pp. 519-531. https://doi.org/10.1007/978-981-13-6001-5_43
[31] Odim, M.O., Ojo, S.O., Oyenike, B. (2023). Analysis of K-Nearest Neighbor for network intrusion detection. Behaviour, 11(5): 52-58. https://doi.org/10.26821/IJSHRE.11.5.2023.110508
[32] Maliha, M. (2021). A supervised learning approach: Detection of cyber attacks. In 2021 IEEE International Conference on Telecommunications and Photonics (ICTP), Dhaka, Bangladesh, pp. 1-5. https://doi.org/10.1109/ICTP53732.2021.9744169
[33] Heramil, J.A., Dumbrique, K., Mirarza, M.R., Ejorango, L.K., Gardon, R.W., Rabago, L. (2023). Threatlocke: An anomaly based detection model. In 2023 8th International Conference on Information Technology and Digital Applications (ICITDA), Yogyakarta, Indonesia, pp. 1-6. https://doi.org/10.1109/ICITDA60835.2023.10426915
[34] Xiao, L., Wan, X., Lu, X., Zhang, Y., Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Processing Magazine, 35(5): 41-49. https://doi.org/10.1109/MSP.2018.2825478
[35] Wardana, A.A., Kołaczek, G., Warzyński, A., Sukarno, P. (2024). Ensemble averaging deep neural network for botnet detection in heterogeneous Internet of Things devices. Scientific Reports, 14(1): 3878. https://doi.org/10.1038/s41598-024-54438-6
[36] Awan, K.A., Din, I.U., Almogren, A., Kim, B.S., Guizani, M. (2024). Enhancing IoT security with trust management using ensemble XGBoost and AdaBoost Techniques. IEEE Access, 12: 116609-116621. https://doi.org/10.1109/ACCESS.2024.3413600
[37] Kim, J., Shim, M., Hong, S., Shin, Y., Choi, E. (2020). Intelligent detection of IoT botnets using machine learning and deep learning. Applied Sciences, 10(19): 7009. https://doi.org/10.3390/app10197009
[38] Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y. (2018). N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3): 12-22. https://doi.org/10.1109/MPRV.2018.03367731
[39] Rathod, G., Sabnis, V., Jain, J.K. (2024). Improving IoT Botnet attack detection using machine learning: Comparative analysis of feature selection methods and classifiers in intrusion detection systems. In 2024 3rd International Conference for Innovation in Technology (INOCON), Bangalore, India, pp. 1-8. https://doi.org/10.1109/INOCON60754.2024.10511883
[40] Seong, T.B., Ponnusamy, V., Jhanjhi, N.Z., Annur, R., Talib, M.N. (2021). A comparative analysis on traditional wired datasets and the need for wireless datasets for IoT wireless intrusion detection. Indonesian Journal of Electrical Engineering and Computer Science, 22(2): 1165-1176. https://doi.org/10.11591/ijeecs.v22.i2.pp1165-1176
[41] Alazzam, H., Alsmady, A., Shorman, A.A. (2019). Supervised detection of IoT botnet attacks. In Proceedings of the Second International Conference on Data Science, E-Learning and Information Systems, Dubai, United Arab Emirates, pp. 1-6. https://doi.org/10.1145/3368691.3368733
[42] Qureshi, S., He, J., Tunio, S., Zhu, N., et al. (2021). A hybrid DL-based detection mechanism for cyber threats in secure networks. IEEE Access, 9: 73938-73947. https://doi.org/10.1109/ACCESS.2021.3081069
[43] Taher, F., Abdel-Salam, M., Elhoseny, M., El-Hasnony, I.M. (2023). Reliable machine learning model for IIoT botnet detection. IEEE Access, 11: 49319-49336. https://doi.org/10.1109/ACCESS.2023.3253432
[44] Almuqren, L., Alqahtani, H., Aljameel, S.S., Salama, A.S., Yaseen, I., Alneil, A.A. (2023). Hybrid metaheuristics with machine learning based botnet detection in cloud assisted internet of things environment. IEEE Access, 11: 115668-115676. https://doi.org/10.1109/ACCESS.2023.3322369