Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/73/20240373

A comparative analysis of feature selection for IoT device identification in machine learning algorithms

Zhiming Guo

School of Computing and Data Science,Xiamen University Malaysia,Kuala Lumpur, Malaysia,43900.

CST2009119@xmu.edu.my

Abstract. In recent years, IoT security has become a critical concern, prompting the integration of artificial intelligence and cybersecurity. State-of-the-art research has focused on applying machine learning and deep learning techniques, emphasizing the crucial role of data preprocessing for effective results. One of the most common and significant data preparation techniques, feature selection is now an essential step in the machine learning process. It involves finding pertinent characteristics and deleting unimportant, low-correlation, redundant, or noisy data. In people's daily lives, noise can interfere with the sound of our normal lives. Similarly, in machine learning, they can also interfere with the process of correctly recognizing the laws of things, leading to the influence of many parameters such as learning rate. This feature selection process improves predictive accuracy and increases comprehensibility. This paper evaluates the performance of four machine learning algorithms for embedded feature selection and compares them. In the end, it was found that XGBoost outperformed the other three algorithms in terms of performance. After feature selection, the accuracy remained almost unchanged, and even with only a few features (about 10% of the original data), it still had a high accuracy

Keywords: Internet of Things, IoT device identification, machine learning, feature selection algorithms, performance comparison.

1. Introduction

With the rapid population growth and development of modern technology. There have been massive computers (including pc and tremendous new computers), internet applications, the rise of the Internet of things and cloud computing, which have generated large and increasing amounts of high-dimensional and complex data such as high-definition video and photo, financial market data, medical imaging data and etc. The interconnection properties of IoT devices studied in this article, combined with their diverse data sources and communication protocols, belong to high-dimensional complex data. These data often have the characteristics of high dimensions, and it makes it difficult for machine learning to process large amounts of this kind of data [1]. Data preprocessing feature selection is now particularly important [2].

Feature selection refers to selecting N features from the existing M features according to a certain standard to optimize the indicators we need to achieve, obtaining an optimal learning sample, which is beneficial for subsequent machine learning. There is a similar concept: feature extraction, which is different from feature selection. It refers to the use of existing features to calculate a more abstract

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/73/20240373

feature set and also refers to the algorithm that calculates a certain feature [3]. Feature selection plays an important role in condensing the data processing scale, where the redundant and irrelevant features will be removed. Good selection results can improve learning accuracy, reduce learning time to improve efficiency, and simplify learning results to facilitate understanding.

Furthermore, within the realm of feature selection for the Internet of Things (IoT), researchers in this domain have illuminated novel avenues that promise to enhance the efficiency of IoT device recognition. Efficiency, in this context, is primarily characterized by two paramount factors: accuracy and time consumption. The choice between prioritizing these factors can be strategically determined through the development of specialized algorithms. For instance, when emphasizing precision, as assessed by accuracy rates, among the four algorithms examined in this study, XGBoost emerges as a judicious selection. Conversely, in scenarios where expedited results are imperative without compromising accuracy, neural network algorithms become a compelling choice. In cases where temporal constraints are stringent, the Support Vector Machine (SVM) algorithm offers an optimal solution.

Moving forward, this research endeavors to conduct in-depth investigations into a wider spectrum of algorithms, aligning each with specific efficiency imperatives, to advance the state-of-the-art in IoT device recognition.

2. Related work

A current study area is the use of machine learning for device identification in the IoT environment. This section reviews the relevant literature, much of which has recently been published. The work connected to IoT device identification is presented first, and the work related to the feature selection approach is reviewed in the second part.

2.1. IoT Device Identification

Research in IoT device identification methods focuses on how to classify and identify different devices based on the data generated by these devices. Traditional classification algorithms such as Support Vector Machine (SVM) and Logistic Regression have found widespread applications in this field. SVM achieves classification by constructing an optimal separating hyperplane that maps device data into a higher-dimensional space, while Logistic Regression makes classification decisions by establishing a probability model. In addition, neural networks and deep learning technologies such as Convolutional Neural Networks (CNNs) and Multi-Layer Perceptrons (MLPs) have also become powerful tools for IoT device identification. These methods can automatically learn feature representations from data, making them suitable for complex device recognition tasks.

With the use of machine learning techniques, Fereshteh Abbasi et al. tackled the issue of anomaly detection and suggested two feature extraction and classification methods. These methods include feature extraction, Logistic Regression (LR)-based classification, and Artificial Neural Networks (ANN)-based classification. The N_BaIoT dataset, which includes data samples linked to nine IoT devices and several attack scenarios, was used for the evaluation. Simulated findings showed that the strategy using Logistic Regression was more successful, reaching the greatest classification accuracy at about 99.98% when compared to four other deep learning approaches based on several assessment parameters, including F1 score and accuracy [4].

2.2. Feature Selection

R. Pavaiyarkarasi et al. have devised a framework for selecting pertinent features in the Internet of Things (IoT) network to facilitate reliable identification of malicious traffic. Following the development and implementation of a novel feature selection metric reliant on wrapper methods for precise feature filtration, they employ the Random Forest algorithm for detecting harmful traffic within IoT networks, aimed at identifying malicious flows. Utilizing the Robot IoT dataset as a foundation, they conduct an evaluation of the efficacy of this strategy. The scrutiny of the exploratory findings substantiates the efficiency of their proposed approach, yielding results exceeding 96 percent accuracy in the majority of cases [5].

DOI: 10.54254/2755-2721/73/20240373

Mnahi Alqahtani et al. presented an effective strategy for identifying IoT botnet assaults. Their methodology utilizes the Hereditary Outrageous Slope Upgrade (GXGBoost) model and component determination strategies in light of Fisher scores. The most important characteristics are unequivocally found utilizing this mix, and IoT botnet attacks are accurately recognized. A vital part of this procedure is the Fisher score, a notable channel based include determination strategy. By filtering a large number of features, reducing intra-class distance and increasing inter-class distance, it successfully separates relevant characteristics from unnecessary features. Furthermore, the GXGBoost model, which is responsible for classifying IoT botnet attacks, is eminent for its viability and optimality. The proposed strategy has shown momentous outcomes in thorough tests condu. Inside and out tests utilizing freely open botnet datasets, including IoT gadgets, have shown the astounding adequacy of the proposed method. Utilizing just three out of 115 information traffic attributes, it accomplishes significant recognition rates by consolidating protection and 10x cross-approval draws near. Therefore, IoT botnet assault location performs for the most part considerably better [6]. In the meantime, XGBoost is likewise one of the calculations we are assessing this time.

3. Overview of IoT Device Identification Algorithms (Linear SVM, Multi Classification Logistic Regression, Neural Network, and XGBoost)

3.1. Linear SVM

Support Vector Machines (SVM) are powerful supervised learning models widely employed in classification and regression tasks. Their core objective is to divide samples into different categories by seeking an optimal hyperplane.

In SVM, samples are represented as points in feature space and assigned to distinct classes. The fundamental concept behind SVM is to find a hyperplane that maximizes the margin (also known as the gap) between different categories. This hyperplane is chosen as the best decision boundary to effectively separate samples of different classes. During the training process, SVM identifies support vectors, which are the samples closest to the hyperplane. These support vectors are crucial for defining the hyperplane and classification boundaries. SVM constructs the model based on these support vectors while treating other samples as noise or of lesser importance [7].

Linear SVM is particularly effective when dealing with linearly separable data, where two categories can be clearly separated by a straight line (or hyperplane). It has robustness to outliers and can effectively handle large-scale features [8].

However, linear SVM may perform poorly on complex nonlinear separable data, in which case nonlinear classifiers such as kernel SVM may be more suitable. Nevertheless, linear SVM is still a popular choice in various application fields, including text classification, image recognition, and sentiment analysis [9].

3.2. Multi-Classification Logistic Regression

A statistical learning technique called logistic regression is applied to solve categorization issues. Although the word "regression" appears in the method's name, it is really a classification algorithm that predicts the results of discrete categories.

Combining a linear model with a logistic function, sometimes referred to as the Sigmoid function, is the basic idea behind logistic regression. Combining these two results in a model that can convert input information into probabilistic outputs [10]. The possibility that a certain sample belongs to a particular class is represented by this probability. The results of logistic regression are between 0 and 1. The sample is typically assigned to the positive class (1) when the output is larger than or equal to 0.5; otherwise, it is assigned to the negative class (0).

The training process of logistic regression involves fitting model parameters through Maximum Likelihood Estimation (MLE) to best fit the observed training data. The model's parameters comprise weights and bias terms, and these parameters are used to calculate the probability of a sample belonging to the positive class [11].

DOI: 10.54254/2755-2721/73/20240373

Logistic Regression is widely employed for binary classification problems, such as spam email classification, disease diagnosis, and customer churn prediction. It can also be extended to multiclass classification problems, where it handles multiple classes through strategies like One-vs-Rest or One-vs-One [12].

The advantages of logistic regression include simplicity, ease of interpretation, and high computational efficiency, making it particularly suitable for linearly separable or approximately linearly separable problems. However, for complex nonlinear problems, the performance of logistic regression may be limited. In such cases, more complex models such as support vector machines or deep neural networks may be more appropriate [13].

3.3. Neural Network

Neural network is a machine learning model inspired by the structure of the human brain. It is widely used to address a variety of complex problems, particularly achieving significant accomplishments in fields such as image recognition, natural language processing, and pattern recognition. Neural networks consist of a large number of interconnected nodes, also known as neurons, which simulate the interactions between biological neurons [14].

Artificial neural networks rely on various related units or centers, known as pseudo neurons, which can effortlessly mimic neurons in the real brain. Every association, like neurotransmitters in the organic brain, has the ability to send signals to different neurons. Pseudo neurons process signals and have the ability to label neurons associated with them. The "signal" related to correlation is a real number, and the result of each neuron is determined by the indirect capacity of its feedback bits. These associations are called edges. The weight of neurons and edges often changes during the learning process. Weights increase or decrease the strength of associative symbols. If the entire information crosses the boundary, neurons may have an advantage in conveying symbols [15].

Neural networks typically consist of multiple layers, including the input layer, hidden layers, and the output layer. The input layer receives raw data, the hidden layers perform feature extraction and representation learning, and the output layer generates the final predictions of the model. Through the backpropagation algorithm, neural networks can adjust their weights based on the error in the model's output, continuously improving the accuracy of the mode [16].

The power of neural networks lies in their ability to automatically discover complex patterns and features within data, making them widely applicable across various domains. These domains include computer vision, speech recognition, natural language processing, medical diagnostics, and more.

3.4. XGBoost

XGBoost is a widely recognized and powerful open-source implementation of the gradient boosting algorithm. Gradient boosting is a supervised learning algorithm that aims to accurately predict a target variable by combining the predictions of a set of simpler, weaker models. In other words, XGBoost builds an ensemble of decision trees sequentially, where each subsequent tree corrects the mistakes made by the previous trees. It assigns weights or importance to each training instance and focuses on the instances that were previously misclassified, allowing the model to learn from those mistakes and improve its predictions [17].

In gradient boosting for regression, regression trees serve as weak learners, where each data point is assigned to a leaf node that holds a continuous score. By combining a smooth loss function (based on the difference between the predicted and target outputs) with a regularization term for model complexity (i.e., the capabilities of regression trees), XGBoost achieves a regularized (L1 and L2) objective function. During the training process, new trees are added that compute the residuals or errors of previous trees. These new trees are then combined with the earlier trees to obtain the final prediction. This iterative training cycle, where new models are introduced by calculating residuals and minimizing loss via a gradient descent approach, is the essence of gradient boosting [18].

Model compression techniques can be used to reduce the complexity of XGBoost models while maintaining their performance and interpretability. One approach is to convert the XGBoost model into

DOI: 10.54254/2755-2721/73/20240373

a single "surrogate" decision tree that approximates the original model's decision function. This allows for a simpler and more interpretable representation of the model without sacrificing its predictive accuracy [19].

4. Introduction of Dataset and Experimental Setup

4.1. Dataset

This study utilized the 'IOT device identification' dataset from the Kaggle machine learning repository [20]. The dataset comprises 1,900 samples, each characterized by 297 features such as 'ack,' 'ack_A,' 'ack_B,' 'byte,' and more. These samples originate from ten distinct categories of Internet of Things (IoT) devices, including security cameras, TVs, smoke detectors, thermostats, water sensors, watches, baby monitors, motion sensors, lights, and sockets. The primary objective of this dataset is to classify these IoT devices based on their features. Widely used in the field of machine learning, it serves as a common example for classification tasks.

The original dataset consists of both a training set and a test set. Due to an uneven allocation ratio of 10:9 between the training and test sets, these two datasets have been combined. Subsequently, the combined dataset was split using the `train_test_split` function, resulting in a training set of 1,520 samples and a test set of 380 samples, maintaining a 4:1 ratio. This division facilitates model training and performance evaluation.

4.2. Experimental Setup

In terms of data Preprocessing, the dataset is imported firstly, and a dictionary (mapping) is created to map the names of IoT device categories to numerical identifiers. Subsequently, using this created mapping, the device category labels in the dataset are replaced with the corresponding numerical identifiers. Furthermore, all missing entries are replaced with zeros. Since all features have already been numericized, there is no need for additional transformations. Columns with zero variance, where the difference between the maximum and minimum values is zero, are directly removed. This step is taken because features with zero variance provide no discriminatory value for machine learning in distinguishing between different devices.

Following this, the target variable, which consists of the numerical device identifiers, and the feature columns are separated and assigned to variables y and x, respectively. Using the `train_test_split` function the dataset is then split into training and testing sets, resulting in x_train, x_test, y_train, and y_test.

The final step before conducting the experiment is the standardization of the data, using the following formula: $x' = (x-\mu)/\sigma$, where μ is the expectation (mean) of the feature and σ is the standard deviation. This process ensures that the data is transformed into a common scale or format, allowing for meaningful comparisons and analyses in the experiment.

Next, we will employ a Support Vector Machine (SVM) classifier with a linear kernel for machine learning classification. Subsequently, one of the metrics for evaluating the classification model's performance, namely Accuracy, will be utilized to assess the model's effectiveness. Additionally, we will analyze the importance of the pre-processed features to determine their contributions to the model's performance.

To begin with, we will train the SVM classifier with a linear kernel using all features and evaluate its performance based on accuracy. Following that, we will rank the features according to their weights and select the top 30, 120, 150, and 200 features. We will then re-run the SVM classifier to observe how different feature quantities affect classification accuracy.

This experiment will aid in comprehending the role of varying feature quantities in machine learning algorithms and how to optimize model performance. Furthermore, we will replicate the same process in subsequent experiments to comprehensively evaluate the performance of the other three machine learning algorithms (Logistic Regression, Neural Networks, and XGBoost).

DOI: 10.54254/2755-2721/73/20240373

5. Performance Evaluation and Result Discussion

In this section, the study will discuss the experimental results and performance evaluation in detail, and conduct a comprehensive analysis and discussion on my Internet of Things device recognition system and embedded feature selection. The experiment first reviewed the performance of each machine learning algorithm, including SVM, logical regression, neural network, and XGBoost. Then, the survey discussed the impact of different feature selection strategies on model performance and discussed the importance of feature selection. Finally, the research discussed the potential opportunities and possible improvement directions for model performance improvement and feature selection.

5.1. Performance evaluation of machine learning algorithms and embedded feature selection methods

In this study, the research used four different machine learning algorithms, namely SVM, logical regression, neural network and XGBoost, to identify IoT devices. Each algorithm has undergone rigorous experimental evaluation, including repeated experiments to evaluate its performance.

The experimental results show that XGBoost performs well, showing significant advantages in accuracy and generalization performance. SVM and logistic regression also show good performance, but in some cases, they may be more sensitive to feature selection of data. The neural network shows good performance in some cases, but its performance is highly dependent on the selection of network structure and the adjustment of super parameters (in the test, I adjusted the number of layers and nodes several times, and finally selected four hidden layers with high accuracy, the number of nodes is 170, 200, 200, and 150 respectively)

The investigation have conducted extensive experiments on feature selection strategies, including sorting based on feature weights and selecting different numbers of feature subsets. Experimental results show that feature selection has a significant impact on model performance. By selecting the most informative feature subset, it can improve the performance of the model, reduce the computational complexity, and maintain high accuracy. Among them, XGBoost has the highest accuracy for datasets with fewer features and lower dimensions, and its accuracy remains almost unchanged after significant dimension reduction.

The original dataset has 297 dimensions, or 297 characteristic columns.

After running machine learning, select the features with the top 200 and 150 weights and run the machine learning algorithm again to calculate accuracy:

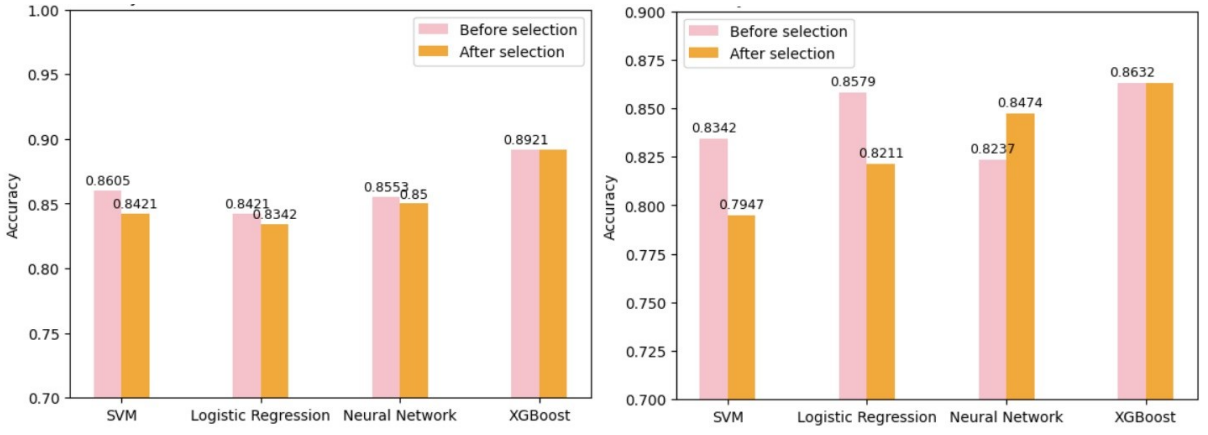

Figure 5. Accuracy Before and After Feature Selection (200 features left and 150 features left)

After running machine learning, select the features with the top 120 and 30 weights and run the machine learning algorithm again to calculate accuracy:

DOI: 10.54254/2755-2721/73/20240373

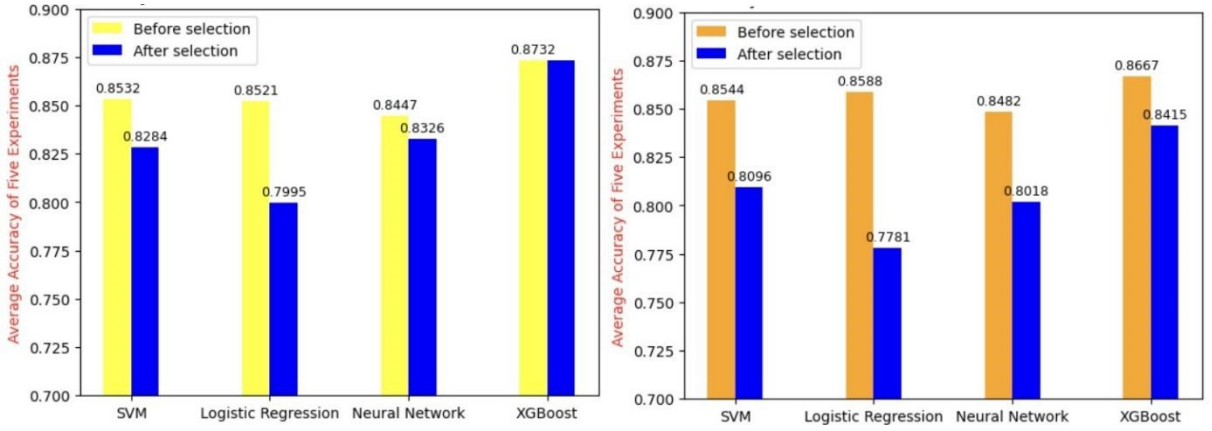

Figure 6. Accuracy Before and After Feature Selection (120 features left and 30 features left)

According to the experimental results, XGBoost still maintains the highest performance among the four algorithms after significant feature selection (dimensionality reduction processing), with an accuracy of 84%.

5.2. Performance Enhancement and Improvement Opportunities

Lastly, the analysis discuss opportunities for further enhancing model performance and improvements. These include refining feature engineering, tuning model parameters (such as adjusting neural network-related parameters), exploring ensemble methods, and gathering more high-quality data. We believe that through continued research and experimentation, it is possible to further improve the performance of the Internet of Things (IoT) device recognition system through feature selection, better meeting the demands of real-world applications.

6. Impact, Limitations, and Future Research Directions

6.1. Impact

This study has undertaken an in-depth exploration of algorithms for IoT device recognition and embedded feature selection methods, yielding a series of meaningful accomplishments that hold positive implications for the field of IoT.

Firstly, a novel IoT device recognition framework based on post-machine learning feature selection was proposed, offering fresh insights and methodologies to address real-world device management and security challenges. This research provides a feasible solution for feature selection in device recognition within IoT application scenarios.

Secondly, through extensive experimental assessments of various machine learning algorithms and feature selection strategies, robust evidence has been furnished for optimizing IoT device recognition tasks. These research findings proffer valuable recommendations to practitioners and researchers regarding the selection of appropriate algorithms and feature subsets.

Finally, for researchers related to the Internet of Things and feature selection, theoretically, this provides us with a new approach to improve the efficiency of IoT device recognition. The two most important factors related to efficiency are accuracy and time spent. We can develop specialized algorithms to accurately calculate which algorithm we need when we prioritize which side, such as accuracy, Among the four algorithms tested in this research institute, we can choose XGBoost; Neural networks can be chosen as they require both shorter time and higher accuracy.

DOI: 10.54254/2755-2721/73/20240373

6.2. Limitations

Notwithstanding the significant achievements of this study, several limitations warrant consideration. Firstly, the experiments were conducted within specific datasets and contexts, potentially constraining their applicability to all IoT device recognition problems. Diverse application scenarios and datasets may necessitate distinct methods and algorithms.

Secondly, feature selection represents a complex issue, and while this study employed common feature selection methods in experiments, numerous unexplored feature selection techniques and strategies could further enhance model performance.

Lastly, this research predominantly focused on supervised learning methods, whereas IoT device recognition problems may also involve semi-supervised and unsupervised learning scenarios. Subsequent investigations should contemplate these more intricate situations.

6.3. Future Research Directions

Future research endeavors can delve deeper into several domains:

Firstly, further exploration of intricate machine learning algorithms and deep learning models holds the potential to enhance the performance of IoT device recognition systems. For instance, considering the utilization of CNN or RNN for handling time-series data.

Secondly, the exploration of additional feature selection methods, encompassing filter-based feature selection, wrapper-based feature selection, and deep learning-based feature learning, might facilitate the extraction of more informative features.

Lastly, expanding IoT device recognition problems to encompass more complex multi-class classification scenarios can better mirror the intricacies of real-world application scenarios.

In summary, IoT device recognition represents a pivotal and challenging issue, and future research will persist in exploring innovative methodologies and solutions to meet the evolving requirements of IoT applications.

7. Conclusion

In this study, I conducted an in-depth investigation into the key issues of IoT device recognition within the context of machine learning. I proposed and evaluated a machine learning-based feature selection framework for IoT device recognition, designed to address practical device management and security concerns. Through extensive experimental evaluations, I conducted a thorough analysis of the performance of various machine learning algorithms and feature selection strategies. It was revealed that the XGBoost algorithm demonstrated a relatively significant advantage in this regard, offering practical insights for IoT device recognition.

The outcomes of this study have a positive impact on the field of IoT. Firstly, I presented a viable solution, which involves employing XGBoost for embedded feature selection followed by the execution of the algorithm to recognize distinct IoT devices. This approach contributes to enhancing the accuracy and efficiency of IoT device recognition. Secondly, my research findings can guide practitioners and researchers in making informed decisions regarding algorithm selection and feature subset choice, ultimately optimizing the performance of IoT device recognition tasks.

However, it is important and necessary to acknowledge the limitations of this study. Particularly, the experiments were conducted within specific datasets and contexts, potentially limiting their generalizability to all IoT device recognition problems. Additionally, the complex issue of feature selection may require further research to enhance model performance. Future research can explore more complex machine learning algorithms and feature selection methods to meet the evolving demands of IoT applications.

In summary, IoT device recognition presents a challenging problem, and this study has provided valuable insights and methods to address it. I anticipate that future research will continue to drive advancements in this field to meet the growing demands of IoT device recognition.

DOI: 10.54254/2755-2721/73/20240373

References

[1] Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70-79.

[2] Kumar, V., & Minz, S. (2014). Feature selection: a literature review. SmartCR, 4(3), 211-229.

[3] Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. In Feature extraction: foundations and applications (pp. 1-25). Berlin, Heidelberg: Springer Berlin Heidelberg.

[4] Abbasi, F., Naderan, M., & Alavi, S. E. (2021, May). Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset. In2021 5th International Conference on Internet of Things and Applications (IoT)(pp. 1-7). IEEE.

[5] Pavaiyarkarasi, R., Manimegalai, T., Satheeshkumar, S., Dhivya, K., & Ramkumar, G. (2022, April). A Productive Feature Selection Criterion for Bot-IoT Recognition based on Random Forest Algorithm. In2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)(pp. 539-545). IEEE.

[6] Alqahtani, M., Mathkour, H., & Ben Ismail, M. M. (2020). IoT botnet attack detection based on optimized extreme gradient boosting and feature selection.Sensors,20(21), 6336.

[7] Schuldt, C., Laptev, I., & Caputo, B. (2004, August). Recognizing human actions: a local SVM approach. InProceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.(Vol. 3, pp. 32-36). IEEE.

[8] Chauhan, V. K., Dahiya, K., & Sharma, A. (2019). Problem formulations and solvers in linear SVM: a review.Artificial Intelligence Review,52(2), 803-855.

[9] Chang, Y. W., & Lin, C. J. (2008, December). Feature ranking using linear SVM. InCausation and prediction challenge(pp. 53-64). PMLR.

[10] Nasteski, V. (2017). An overview of the supervised machine learning methods.Horizons. b,4, 51-62.

[11] Indra, S. T., Wikarsa, L., & Turang, R. (2016, October). Using logistic regression method to classify tweets into the selected topics. In2016 international conference on advanced computer science and information systems (icacsis)(pp. 385-390). IEEE.

[12] Alhessi, Y., & Wicentowski, R. (2015, June). Swatac: A sentiment analyzer using one-vs-rest logistic regression. InProceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)(pp. 636-639).

[13] Uncuoglu, E., Citakoglu, H., Latifoglu, L., Bayram, S., Laman, M., Ilkentapar, M., & Oner, A. A. (2022). Comparison of neural network, Gaussian regression, support vector machine, long short-term memory, multi-gene genetic programming, and M5 Trees methods for solving civil engineering problems.Applied Soft Computing,129, 109623.

[14] Hardesty, L. (2017). Explained: neural networks.MIT News,14.

[15] Brahme, A. (2014).Comprehensive biomedical physics. Newnes.

[16] Müller, B., Reinhardt, J., & Strickland, M. T. (1995).Neural networks: an introduction. Springer Science & Business Media.

[17] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. InProceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining(pp. 785-794).

[18] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... & Zhou, T. (2015). Xgboost: extreme gradient boosting.R package version 0.4-2,1(4), 1-4.

[19] Sagi, O., & Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree.Information Sciences,572, 522-542.

[20] Ami (2021). IoT Device Identification [A dataset for identifying IoT devices]. Kaggle. https://www.kaggle.com/datasets/fanbyprinciple/iot-device-identification)