An Efficient Intrusion Detection System for IoT Using Feature Engineering and Machine Learning

Basima Kzar Hassan[1], Khawla Rashige Hassen[2] and Sakna Jahiya Faraj[3]
[1]*Department of Economic Studies, Center for Basra and Arab Gulf Studies, University of Basrah, 61004 Basra, Iraq*
[2]*Department of Field Crops, College of Agriculture, University of Basrah, 61004 Basra, Iraq*
[3]*Department of Economics, College of Economics and Administration, University of Basrah, 61004 Basra, Iraq*
basima.hasan@uobasrah.edu.iq, khawla.hassan@uobasrah.edu.iq, sakna.al-sary@uobasrah.edu.iq

Keywords: Intrusion Detection System (IDS), Internet of Things (IoT), Machine Learning, Feature Engineering, SMOTE.

Abstract: IoT devices have limited computing and security capabilities, making them vulnerable to cyberattacks. This rapid expansion of IoT has introduced unprecedented connectivity but also heightened security vulnerabilities. The security of IoT environments, therefore, depends on efficient and lightweight intrusion detection systems (IDS). Using advanced feature engineering and machine learning algorithms, this study develops high-performance IDS designed for IoT networks. Data imbalance is addressed with preprocessing techniques, feature extraction, and synthetic minority oversampling techniques (SMOTE). Multi-dataset training and testing included K-nearest neighbour models, sequential minimalism optimization models, random forest models, and stacking ensembles. A transfer learning model such as VGG-16 and DenseNet was also incorporated to improve classification accuracy. It has been demonstrated that the proposed models, particularly the ensemble and RF-based approaches, are highly accurate, precise, and recallable. IoT environments with limited resources can benefit from the proposed IDS framework because it effectively identifies malicious traffic while maintaining computational efficiency.

1 INTRODUCTION

Internet-connected devices are called IoT devices. Things such as wireless sensors, smart cameras, and smart televisions are being connected to the IoT, which has undergone rapid changes [1], [2]. IoT devices are rapidly spreading across the Internet, with more than 2 billion connected in 2017. Data generated by Internet-of-things devices is projected to reach 73.1 zettabytes by 2025, according to experts [3]. As the IoT grows in popularity and offers many benefits, their security can often be limited even though IoT generates vast amounts of data and is growing at an unprecedented rate. IoT has been rapidly adopted due to the huge amount of data generated by
billions of connected devices. Even as IoT devices become increasingly popular, they lack strong security protections.

IoT devices lack sufficient security capabilities, so it's essential to build NIDS that can detect and prevent attacks on IoT networks quickly and reliably [4]. As well as machine learning techniques, publicly available data on network traffic can also be used for intrusion detection in IoT [5], [6]. Machine learning models tend to be more complex and accurate when these datasets contain fewer redundant or irrelevant features [7]. Machine learning models are often developed through feature reduction in order to develop efficient NIDS. In addition to reducing computational costs and latency, model generalization is also increased.

The problem of excessive features can be solved using several techniques, including feature selection and feature extraction. Feature selection selects the most informative features from an initial set of features [8]. The selected features remain semantically interpretable while the dimensionality is reduced. By mathematically projecting the original features into a low-dimensional space, feature extraction is performed [9], [10]. There is no intuitive meaning to the extracted features, which may reduce dimensionality, but they may reduce complexity. In the IoT, lightweight and efficient IDS can be easily created by carefully selecting original features that are relevant to the device in question. As an alternative, a feature extraction technique reduces the overall dimensions of the data while preserving critical information. The optimization of the

efficiency and interpretability of intrusion detection models, which are tailored to the limitations and complexity of IoT devices and networks, can help IoT ecosystems improve their cybersecurity posture.

Many sectors are experiencing an increase in cyber threats, such as IoT, online banking, industrial systems, and healthcare. A smart home, a smart city, and even wearables are all made possible by the IoT. There are, however, some limitations to IoT devices operating over public networks, including limited computational power, storage, and bandwidth [11]. Since these limitations limit their attack potential, they are more vulnerable to attacks than traditional endpoint devices. There is room for improvement in intrusion detection techniques, especially when it comes to improving their accuracy and adaptability, regardless of the number of intrusion detection techniques proposed in the literature.

Here are some contributions to the ongoing study that address these challenges:

- A literature review is presented on the use of machine learning and deep learning for intrusion detection based on numerical and image-based datasets.
- To correct for class imbalances, datasets are preprocessed and balanced using the SMOTE.
- The combination of multiple feature extraction techniques with stacked machine learning methods, including KNN, SMO, and RF, is used to identify malicious or benign network traffic.
- Performing experiments to validate the proposed models' performance and reliability.

2 LITERATURE REVIEW

IoT architectures refer to mechanisms for interconnecting addressable electronic gadgets using radio and telecommunication infrastructures in order to establish interconnection [12]. The communication infrastructure additionally includes living life, such as people, animals, and plants, where it is used for monitoring, improving the quality of life, and reducing resource utilization [13]. WSNs anchor IoT. In addition to collecting information, WSNs do a number of other things [14]. Currently, IoT security frameworks and sensor centres are lacking, and gadgets are not protected against attacks such as Denial of Service (DoS), Man-in-the-middle attacks, etc [15], [16]. IoT verification and access control conventions are currently undergoing a lot of activity, but additional mechanisms are needed according to requirements. Experts have expressed concerns about IoT, its network scheme, and safety concerns [17]. Within the network design architecture, IoT requires consistent data handling and communication.

2.1 Machine Learning-Based Intrusion Detection Systems

In this study, we attempted to determine what set of hyperparameters would be most appropriate for use in NIDS. Data from UNSW-NB15 was analyzed using DFF and LSTM architectures. As compared to DFF, LSTM performed slightly better, but the relu activation function outperformed them all. SGD performed less accurately than the majority of optimisers, except for the majority of them. In their opinion, the best settings for the hyperparameters were relu, adam, and nodes configuring input and output rules at 0.75. DFF had the best accuracy at 98.8%, while LSTM had the best accuracy at 98%. This paper does not drop the flow identifier features, nor does it evaluate their best-claimed set of hyperparameters on a different dataset. As an FE tool, [18] proposed an LSTM-based AE neural architecture composed of dense layers and LSTMs. RF classifiers are applied after extraction to detect attacks. We evaluated the proposed methodology using several datasets, including UNSW-NB15, ToN-IoT, and NSL-KDD [19]. This study indicates that the selected classifier is able to detect more objects without relying on compression to do so. The reduction of dimensions has significantly reduced training time.

In the literature, researchers continue to develop new FR methods and ML models based on the negative habits discussed in [20]. Research in this domain is always able to find combinations or variations to achieve slightly better results numerically. If you are applying the algorithm to a specific dataset, you should modify all hyper-parameters used. Almost all papers have used a single dataset, making it difficult to conclude that their proposed techniques are generalizable. Depending on which dataset you use, the information presented will vary. This means the results of these proposed techniques may vary depending on the dataset to which they are applied. Due to the above experimental issues, ML-based NIDS cannot be deployed in the operational arena despite extensive academic research. Despite this, machine learning tools have been applied successfully to commercial scenarios, compared with other applications. A suitable ML model is necessary before deployment because of NIDS' high error costs. Our best combination will be generalizable if we compare all datasets. Analyzing the extracted dimensions requires

calculating and comparing the variance and correlation between PCA and LDA.

Several evaluation metrics were used in the evaluation of these algorithms, including KDD CUP 99, NSL-KDD, CIDDS, and CICIDS2017. DBN, which enhanced detection accuracy from 5% to 10%, was found to be more effective in DL models than ML models. According to [21], heuristic intrusion detection can achieve an accuracy of 85.5% to 95.2%. Gradient decomposition algorithms are used to train IDSs beforehand, followed by retraining and testing using KDD20+ and KDDTest+ datasets.

By combining NSL-KDD and UNSW-15 datasets, [22] experiments with hybrid sampling-based intrusion detection. With SMOTE and OSS combined, RF, CNN, BiLSTM, CNN-BiLSTM, AlexNet-5, and CNN-BiLSTM models are cross-trained. For the datasets mentioned, CNN-BiLSTM had an accuracy of 83.58% and 77.16%, respectively, outperforming other algorithms. Using RNN and GRNN algorithms, the author investigated the effectiveness of bi-RNN intrusion detection [23]. In the evaluation, Bi-RNN was found to achieve the best accuracy with 99.04% when compared to other methods utilizing 10% KDD datasets [24].

2.2 IoT Threats

As IoT networks are distributed, their architecture is layered, with each layer performing its tasks sequentially to maintain the platform's efficiency. Researchers have found that a variety of attacks can be carried out at every layer of the network for the purpose of breaching its security, including protocols and gateways [25]. There has been a lot of recent attention paid to encapsulated attacks that occur in the whole network, such as DoS, Probe, U2R, and R2L [26].

3 METHODOLOGY

These sections describe the dataset, how it was preprocessed, how features were extracted, and how the dataset was evaluated. Modelling is primarily intended to identify regular traffic versus malicious traffic. This is accomplished by developing several models, representing them in image format, and comparing them. A set of datasets as described in the research paper "Detecting intrusions use network traffic profiling and machine learning for IoT applications using network traffic profiling". The first step is to provide the machine learning module with binary visualization data that is compressed. This repository also provides the packet capture images from each of the five attack scenarios, along with their associated PCAP files, in attackScenario.zip and attackSenarioImages.zip.

3.1 Data Processing

To improve the training process for machine learning models, it is essential to process data first. You can download all datasets for research purposes for free. To reduce storage size and eliminate redundant samples (flows), duplicate samples (flows) are removed. Source and destination IP addresses, ports, and timestamps are stripped from flow identifiers to prevent bias against attackers or victims. A numerical value is encoded for strings, non-numeric features, and other features. In the datasets, protocol and service characteristics are stored in string form, while ML models prefer numerical information. To encode features, one can use hot encoding, while the other can use label encoding. By adding X features to a feature, it becomes X categories. A category represents 1, and absence represents 0. Due to the increased number of dimensions in the dataset, ML models may struggle with performance and efficiency. Thus, each category is assigned a numerical value according to the label encoding technique.

Our dataset consists only of numerical values, eliminating nan, dash, and infinity. A 1 is displayed when a boolean feature is true, while a 0 is displayed when a boolean feature is false. In an effort to reduce complexity, all feature values are scaled between 0 and 1 based on the min-max feature scaling method. The ML model gives equal weight to all features, but since network traffic features tend to have large values, it may give them a heavier weight since they tend to have high values. The values of every feature are calculated by plugging in a new feature value from 0 to 1, the previous feature value, $X*$, and the maximum and minimum of the feature. Datasets for training and testing are separated according to the label features, which is necessary because there is a class imbalance between the two datasets:

$$X* = \frac{X - X_{min}}{X_{max} - X_{min}}. \quad (1)$$

3.2 Synthetic Minority Oversampling Technique (SMOTE)

Using SMOTE [23], balanced datasets are generated by oversampling minority classes. Synthetic examples are included along with the neighbours

nearest to the minority class so that the minority class can be oversampled. If oversampling is necessary, the k-nearest neighbours may be selected at random. This method is illustrated in Figure 1 by Y_i representing the point in question, Y_{i1} through Y_{i4} representing nearest neighbours, and W1 through W4 representing the synthetic data generated by randomized interjection.

Using a feature vector (sample) based on the nearest neighbour, synthetic samples can be generated. The feature vector is calculated by multiplying the difference by a random number between 1 and 0. Using this method, two distinct characteristics are considered when selecting a random point on the line segment. The following is a detailed description of SMOTE:

- Rank samples according to their k-nearest neighbours.
- Samples are selected randomly using the KNN algorithm.
- To calculate the new samples, add the original samples to the difference and then add the gap (0,1).
- Make the minority sample larger by adding new samples. The final step is to create a new dataset.

The oversampling of minority samples by SMOTE has some weaknesses since majority samples are not taken into account when creating minority samples. As a result, minority samples may be generated around the positive examples, escalating the problem of borderline and noisy examples in learning.

3.3 Feature Extraction

3.3.1 Overview of Feature Extraction Process

When a character recognition system is preprocessed, features are extracted. An input pattern must be correctly assigned to one of the possible output classes in order for pattern recognition to succeed. In general, two general stages are involved in this process: Feature selection and classification. Poorly selected features cannot be recognized by the classifier, so feature selection is very important. According to Lippman, the following features should be selected:

Feature information should be sufficient to generate discriminant functions efficiently and be insensitive to irrelevant variations in input, as well as limited in number in order to minimize training data requirements. To build a pattern classification, feature extraction is important for determining the relevant characteristics of each class. As part of this process, relevant features are extracted to form feature vectors using an object/alphabet. Input units and target output units can be recognized by classifiers when these feature vectors are combined. By considering these features, the classifier can differentiate between classes more easily, making it easier to classify. The process of extracting features from raw data involves retrieving the most relevant information. Identifying the parameters that define characters allows them to be accurately and uniquely defined. A feature vector represents a character's identity in a feature extraction process. When generating features for a variety of instances of a symbol, the goal is to maximize the recognition rate using the simplest elements possible.

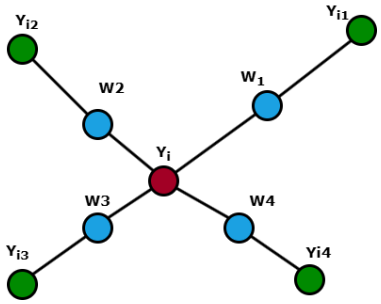

Figure 1: Generate data in the SMOTE algorithm.

3.3.2 Deep Learning-Based Feature Extraction with Pre-Trained Models

Pre-trained models are neural networks that have been trained using a lot of data. Pretrained models include the following. Models that were pre-trained and supervised. It is one of the most popular pre-training datasets for supervised learning. As part of the ImageNet classification challenge, [27] used deep neural networks to outperform humans. InceptionNet [28] is another deep neural network that uses parallel convolutional filters. ResNet [29] introduces skip connections to ease training and becomes much deeper as performance improves. DenseNet [30] densely connected blocks [31] are carefully designed to be mobile-friendly, and the structure can be further optimized through network architecture searches [32]. DenseNet, ResNet_50, and VGG-16 were used. A visibility evaluation model was constructed using the extracted features, and the variables were input into a Support Vector Regression model, followed by a visibility evaluation of the effective area:

- ImageNet was used to train the VGG-16 network [33], [34]. A large training set enables

the VGG-16 to perform well even with small image data sets. There are 16 convolution layers in the VGG-16 network, along with a 3x3 receptive field. The pooling layer, which has a size of 2x2, is one of five. Immediately following the last layer of Max pooling are three fully connected layers. There are three fully interconnected layers in the following layers. The Softmax classifier is the final layer of the algorithm. There are no hidden layers that are not activated by ReLu [35].

- The DenseNet is an artificial neural network based on dense convolutions [36]. A DenseNet architecture consists of interconnected layers. The number of layers in a network is N(N+1)/2. DenseNet comprises an initial layer of convolutions and a layer of pooling. Pre-trained models are neural networks that have been trained using a lot of data.

3.4 Image Filter

With an image filter (IF), high-quality images are removed from highly corrupted images by removing impulse noise from additive identical independent distributions (i.i.d.). To implement the proposed filter, fuzzy numbers need to be constructed, fuzzy filtering needs to be performed, and a genetic learning process needs to be applied. Fuzzy numbers are constructed by constructing the image knowledge base from sample images or noise-free images. Additionally, fuzzy decision processes, fuzzy means, and fuzzy inference mechanisms are all part of fuzzy filtering. The optimization of image knowledge bases through genetic algorithms is finalized by genetic learning. As a method of removing additive impulse noise from highly corrupted images, we will use the genetic learning method to tune the parameters of membership functions.

3.4.1 Auto-Color Correlogram

For optimal accuracy, a number of models were evaluated using a variety of filters, individual learning algorithms, and stacked models. Autocorrelograms were used to develop the first four models. A KNN model, an SMO model, an RF model, and other algorithms were used. Stacking KNN and SMO models was also created.

3.4.2 Fuzzy Color and Texture Histogram

Fuzzy Color and Texture Histogram (FCTH) is a quantitative histogram that includes information about colour and texture [8]. In this feature result, three fuzzy units have been combined. Segmenting the image into blocks is done initially. Every block passes through the fuzzy units. The first unit involves extracting the fuzzy linking histogram using fuzzy rules. Based on the HSV colour space, this histogram can be viewed. A fuzzy system with three inputs generates a 10-bin histogram using twenty rules; each bin corresponds to a colour.

3.5 Performance Measures and Validation

Evaluation of classifiers is typically based on calculating fundamental performance metrics. In this study, the classifier's effectiveness was measured using key statistical values including true positives, true negatives, false positives, and false negatives. From these, important performance indicators such as accuracy, precision, recall, and the F1-score were derived to comprehensively assess the model's predictive capabilities. These metrics provide insight into the classifier's correctness, reliability, sensitivity, and balanced performance. The detailed methodology for calculating these metrics follows the framework established by [37].

4 RESULTS AND DISCUSSION

As shown in Figure 2, four models are compared for accuracy, precision, recall, and F1-score. According to this example, the K-Nearest Neighbor (KNN) model with k = 1 provides the best accuracy (93%), precision (91.4%) and test coverage (91%).

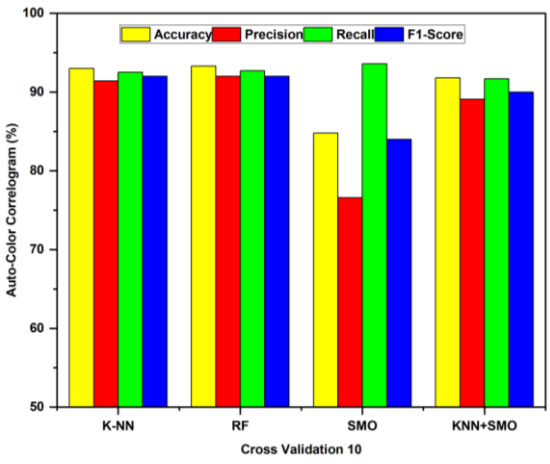

Figure 2: Performance comparison of classification models based on four methods using 10-fold cross-validation.

We evaluated five models, and Figure 3 shows their accuracy results. On the 90-10 dataset, the Random Forest (RF) model was 94.6% accurate and 94% precise, whereas the stacked model was 94.6% accurate and 94% precise. However, Random Forests outperform stacked models when it comes to recall, achieving 100% as opposed to 94.7% for stacked models. The Random Forest model remains accurate with 95% accuracy even when data is split 70/30.
As shown in Figure 4, the accuracy results of each model are summarized. As a result of 90% training and 10% testing, the RF model yielded the highest accuracy and precision, respectively, of 94.22% and 91.02%.

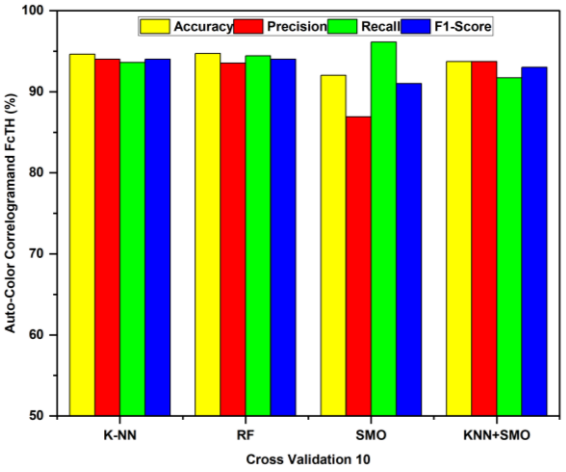

Figure 3: Performance comparison of classification models on FCTH features using 10-fold cross-validation.

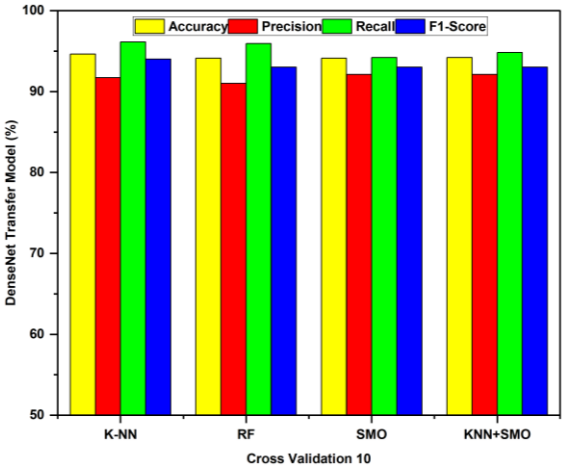

Figure 4: Comparison of four models with DenseNet transfer model.

A comparison of the four models is presented in Figure 5. When 90% of the training data are combined with 10% of the testing data, a model's accuracy and precision are 94.22% and 92.42%, respectively. With 70/30 training data, the KNN model with k = 5 shows 95.8% accuracy and 95.1% precision.

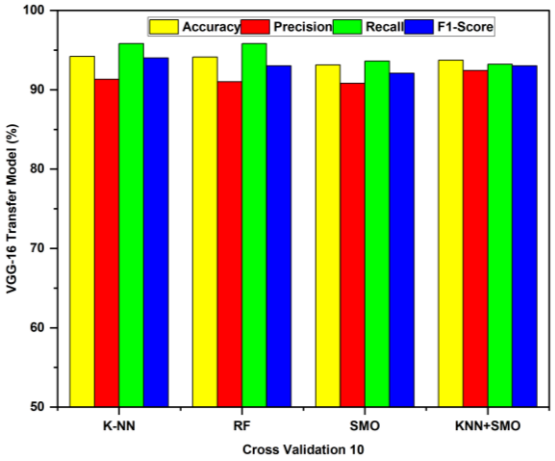

Figure 5: Comparison of four models with VGG-16 transfer model.

5 CONCLUSIONS

An IDS for the IoT is developed by integrating feature engineering techniques with traditional and advanced machine learning models. By effectively preprocessing data, performing thorough feature analysis, and utilizing SMOTE for handling class imbalance, the system significantly enhances the accuracy of intrusion detection across diverse attack scenarios in IoT networks. Multiple classifiers, such as Random Forest, K-Nearest Neighbors (KNN), Sequential Minimal Optimization (SMO), and various ensemble-based methods, exhibited commendable performance, with Random Forest and ensemble approaches consistently outperforming others in terms of accuracy, sensitivity, and precision metrics. The robustness and adaptability of the proposed system were further improved by incorporating image-based feature representations and employing advanced transfer learning techniques using pre-trained models like VGG-16 and DenseNet. These deep learning strategies effectively captured complex patterns within the data, providing superior generalization capabilities. Consequently, the proposed framework delivers a practical, reliable, and scalable solution suitable for real-time threat

detection in resource-constrained IoT environments. Future advancements in IDS applications can greatly benefit from extending this framework towards real-time deployment scenarios and integrating federated learning methods to enhance data privacy, decentralization, and collaborative learning across distributed IoT devices.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015, [Online]. Available: https://doi.org/10.1109/COMST.2015.2444095.

[2] B. Bhola et al., "Quality-enabled decentralized dynamic IoT platform with scalable resources integration," IET Communications, 2022.

[3] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1606–1616, Apr. 2019, [Online]. Available: https://doi.org/10.1109/JIOT.2018.2847733.

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019, [Online]. Available: https://doi.org/10.1109/COMST.2019.2896380.

[5] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 686–728, 2019, [Online]. Available: https://doi.org/10.1109/COMST.2018.2847722.

[6] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," Computers & Electrical Engineering, vol. 105, p. 108543, 2023.

[7] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System," IEEE Access, vol. 7, pp. 94497–94507, 2019, [Online]. Available: https://doi.org/10.1109/ACCESS.2019.2928048.

[8] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999, [Online]. Available: https://researchcommons.waikato.ac.nz/bitstream/10289/15043/1/thesis.pdf. [Accessed: Apr. 25, 2025].

[9] B. Yan and G. Han, "Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System," IEEE Access, vol. 6, pp. 41238–41248, 2018, [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2858277.

[10] N. Hussain, P. Rani, H. Chouhan, and U. S. Gaur, "Cyber security and privacy of connected and automated vehicles (CAVs)-based federated learning: challenges, opportunities, and open issues," Federated Learning in IoT Applications, pp. 169–183, 2022.

[11] P. Rani and R. Sharma, "Intelligent Transportation System Performance Analysis of Indoor and Outdoor Internet of Vehicle (IoV) Applications Towards 5G," Tsinghua Science and Technology, vol. 29, no. 6, pp. 1785–1795, Dec. 2024, [Online]. Available: https://doi.org/10.26599/TST.2023.9010119.

[12] A. M. Mhlaba and M. Masinde, "An Integrated Internet of Things Based System for Tracking and Monitoring Assets–the case of the Central University of Technology," in IST-Africa 2015 Conference Proceedings, 2015, [Online]. Available: https://www.academia.edu/download/66891365/An_Integrated_Internet_of_Things_Based_S20210504-29331-1occ0bs.pdf. [Accessed: Apr. 25, 2025].

[13] F. T. First and I. B. Blocks, "How the Internet of Things Is Revolutionizing Healthcare," [Online]. Available: http://euro.ecom.cmu.edu/resources/elibrary/ubi/IOTREVHEALCARWP.pdf. [Accessed: Apr. 25, 2025].

[14] M. Masinde, "IoT applications that work for the African continent: Innovation or adoption?," in 2014 12th IEEE International Conference on Industrial Informatics (INDIN), IEEE, 2014, pp. 633–638, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6945587/. [Accessed: Apr. 25, 2025].

[15] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE, 2015, pp. 336–341, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7412116/. [Accessed: Apr. 25, 2025].

[16] N. Hussain, P. Rani, N. Kumar, and M. G. Chaudhary, "A deep comprehensive research architecture, characteristics, challenges, issues, and benefits of routing protocol for vehicular ad-hoc networks," International Journal of Distributed Systems and Technologies (IJDST), vol. 13, no. 8, pp. 1–23, 2022.

[17] X. Xingmei, Z. Jing, and W. He, "Research on the basic characteristics, the key technologies, the network architecture and security problems of the internet of things," in Proceedings of 2013 3rd International Conference on Computer Science and Network Technology, IEEE, 2013, pp. 825–828, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6967233/. [Accessed: Apr. 25, 2025].

[18] A. Andalib and V. Tabataba Vakili, "A Novel Dimension Reduction Scheme for Intrusion Detection Systems in IoT Environments," arXiv e-Prints, p. arXiv-2007, 2020.

[19] P. Rani et al., "Federated Learning-Based Misbehavior Detection for the 5G-Enabled Internet of Vehicles," IEEE Transactions on Consumer Electronics, vol. 70, no. 2, pp. 4656–4664, May 2024, [Online]. Available: https://doi.org/10.1109/TCE.2023.3328020.

[20] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in 2010 IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 305–316, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5504793/. [Accessed: Apr. 25, 2025].

[21] A.-H. Qureshi, H. Larijani, J. Ahmad, and N. Mtetwa, "A Heuristic Intrusion Detection System for Internet-of-Things (IoT)," in K. Arai et al. (eds), Intelligent Computing, in Advances in Intelligent Systems and Computing, vol. 997, Cham: Springer International Publishing, 2019, pp. 86–98, [Online]. Available: https://doi.org/10.1007/978-3-030-22871-2_7.

[22] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," IEEE Access, vol. 8, pp. 32464–32476, 2020.

[23] A. Dushimimana, T. Tao, R. Kindong, and A. Nishyirimbere, "Bi-directional recurrent neural network for intrusion detection system (IDS) in the internet of things (IoT)," International Journal of Advanced Engineering Research and Science, vol. 7, no. 3, 2020, [Online]. Available: https://www.academia.edu/download/113495300/68IJAERS-03202047-Bi-directional.pdf. [Accessed: Apr. 25, 2025].

[24] G. Ansari, P. Rani, and V. Kumar, "A novel technique of mixed gas identification based on the group method of data handling (GMDH) on time-dependent MOX gas sensor data," in Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022, Springer, 2023, pp. 641–654.

[25] F. Farhin, M. S. Kaiser, and M. Mahmud, "Secured Smart Healthcare System: Blockchain and Bayesian Inference Based Approach," in M. S. Kaiser et al. (eds), Proceedings of International Conference on Trends in Computational and Cognitive Engineering, in Advances in Intelligent Systems and Computing, vol. 1309, Singapore: Springer Singapore, 2021, pp. 455–465, [Online]. Available: https://doi.org/10.1007/978-981-33-4673-4_36.

[26] N. Islam, I. Sultana, and M. S. Rahman, "HKMS-AMI: A Hybrid Key Management Scheme for AMI Secure Communication," in M. S. Kaiser et al. (eds), Proceedings of International Conference on Trends in Computational and Cognitive Engineering, in Advances in Intelligent Systems and Computing, vol. 1309, Singapore: Springer Singapore, 2021, pp. 383–392, [Online]. Available: https://doi.org/10.1007/978-981-33-4673-4_30.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034, [Online]. Available: http://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html. [Accessed: Apr. 19, 2025].

[28] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9, [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html. [Accessed: Apr. 19, 2025].

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html. [Accessed: Apr. 19, 2025].

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html. [Accessed: Apr. 19, 2025].

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.html. [Accessed: Apr. 19, 2025].

[32] M. Tan et al., "Mnasnet: Platform-aware neural architecture search for mobile," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2820–2828, [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Tan_MnasNet_Platform-Aware_Neural_Architecture_Search_for_Mobile_CVPR_2019_paper. [Accessed: Apr. 19, 2025].

[33] S. Pal, "Transfer learning and fine tuning for cross domain image classification with Keras," GitHub: Transfer Learning Fine Tuning Cross Domain Image Classification with Keras, 2016.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5206848/. [Accessed: Apr. 20, 2025].

[35] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," Construction and Building Materials, vol. 157, pp. 322–330, 2017.

[36] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and M. Chowdhury, "A review of deep learning-based detection methods for COVID-19," Computers in Biology and Medicine, vol. 143, p. 105233, 2022.

[37] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” J. Mach. Learn. Technol., vol. 2, pp. 37–63, 2011, doi: 10.9735/2229-3981.