Eksplorium *p*-ISSN 0854-1418
Volume 46 No. 2, June 2025: 1032–1041 *e*-ISSN 2503-426X

Intrusion Detection using Anomaly Detection and Isolation Forest Method for Implementation and SHAP for Interpretability.

P Vamsi Naidu[1*], B Basaveswara Rao[2]

1,2 Acharya Nagarjuna University, Guntur, AP, India.
*E-mail: simhadri.mallikarjun9@gmail.com

Article Received: 08 May 2025, Revised: 13 June 2025, Accepted: 22 June 2025

Abstract

The rapid proliferation of Internet of Things (IoT) devices has increased the risk of cyber intrusions, necessitating robust and intelligent Intrusion Detection Systems (IDS). Traditional IDS methods struggle with the dynamic nature of IoT networks and the growing sophistication of cyberattacks. In this study, we propose a novel framework for anomaly detection in IDS using the Isolation Forest (IF) method on the RT-IoT2022 dataset. The framework leverages unsupervised learning to identify anomalous network behavior and potential cyber threats in real-time. To enhance interpretability, we integrate SHapley Additive Explanations (SHAP) to provide explainable AI insights into the model's decision-making process. The SHAP technique helps in understanding the contribution of individual features towards anomaly detection, thereby improving transparency and trust in the IDS. Our evaluation metrics, including precision, recall, F1-score, and confusion matrix analysis, demonstrate the efficiency of the proposed model in detecting malicious activities. The experimental results validate the effectiveness of our approach in identifying anomalies while ensuring model interpretability, making it a promising solution for securing IoT networks against evolving cyber threats.

Keywords: Anomaly Detection, Intrusion Detection System, Isolation Forest, Explainable AI, SHAP, RT-IoT2022, IoT Security.

1. Introduction

The rapid advancement of the Internet of Things (IoT) has led to an exponential increase in connected devices across various domains, including healthcare, smart homes, industrial automation, and transportation. While this connectivity brings significant benefits, it also introduces substantial security challenges. IoT networks are highly dynamic, heterogeneous, and resource-constrained, making them vulnerable to a wide range of cyber threats, such as Distributed Denial-of-Service (DDoS) attacks, botnets, and unauthorized access. Traditional Intrusion Detection Systems (IDS) often rely on signature-based or rule-based approaches, which struggle to keep pace with the evolving nature of cyberattacks and the vast volume of network traffic generated by IoT devices. These limitations necessitate the development of intelligent, adaptive, and explainable intrusion detection mechanisms.

In recent years, machine learning (ML)-based IDS solutions have gained prominence for their ability to detect previously unseen attacks and adapt to evolving threat landscapes. Among these approaches, anomaly detection techniques have shown promising results in identifying malicious activities by distinguishing normal network behavior from abnormal patterns. Unsupervised learning models, such as the Isolation Forest (IF), have demonstrated high efficacy in detecting anomalies without requiring labeled training data, making them well-suited for IoT security applications.

This research proposes a novel anomaly detection framework utilizing the Isolation Forest method to enhance intrusion detection capabilities in IoT networks. The framework is designed to analyze network traffic data from the RT-IoT2022 dataset and identify potential cyber threats in real time. To address the challenge of model interpretability, we integrate SHapley Additive Explanations (SHAP), an Explainable AI (XAI) technique that provides insights into the model's decision-making process. SHAP helps security analysts and network administrators understand the contribution of individual network features to the anomaly detection process, thereby improving trust, transparency, and practical applicability in real-world cybersecurity settings.

The effectiveness of the proposed framework is evaluated using key performance metrics, including precision, recall, F1-score, and confusion matrix analysis. Experimental results demonstrate that the combination of the Isolation Forest model with SHAP explanations significantly enhances intrusion detection while maintaining interpretability. By providing a robust and explainable anomaly detection system, this study contributes to strengthening the security of IoT networks against emerging cyber threats.

The rest of the paper is structured as follows: Section 2 discusses related work in IoT security and anomaly detection techniques. Section 3 presents the proposed methodology, including data preprocessing, model training, and the integration of SHAP explanations. Section 4 outlines the experimental setup and performance evaluation. Section 5 discusses the results, highlighting key findings and implications. Finally, Section 6 concludes the paper and outlines future research directions.

2. Literature Survey

Several studies have explored ML and DL models for IDS implementation in IoT networks. Shone et al. (2018) proposed a deep learning-based intrusion detection system using a combination of autoencoders and Random Forest classifiers. Their model demonstrated improved detection capabilities for unknown attacks. Similarly, Mirsky et al. (2018) introduced Kitsune, an ensemble-based anomaly detection system leveraging autoencoders to identify network intrusions in real-time. These studies highlight the effectiveness of unsupervised learning in IoT security but lack interpretability, making it difficult to analyze decision-making processes.

More recently, Khan et al. (2020) evaluated various ML models, including Support Vector Machines (SVM), Decision Trees (DT), and Artificial Neural Networks (ANN), for intrusion detection in IoT environments. Their study concluded that hybrid models combining multiple classifiers enhanced attack detection accuracy. However, ML models often face challenges related to high false positive rates and imbalanced datasets, necessitating further research in anomaly detection techniques.

ii. Anomaly Detection in IoT Networks Anomaly detection methods, particularly unsupervised learning models, have gained traction due to their ability to identify previously unknown attacks without labeled data. Isolation Forest (IF), One-Class SVM, and k-means clustering are some of the widely used models for anomaly detection. Liu et al. (2012) introduced the Isolation Forest algorithm, demonstrating its effectiveness in outlier detection by isolating anomalies through recursive partitioning. This method has been successfully applied in cybersecurity applications, including IoT intrusion detection (Yousefi-Azar et al., 2017).

Studies such as those by Yin et al. (2021) have explored the effectiveness of IF for detecting network intrusions in IoT ecosystems. Their research emphasized IF's ability to efficiently isolate anomalous network behavior with minimal computational overhead. However, despite its success, IF-based IDS lacks explainability, making it challenging to understand the rationale behind anomaly classification.

iii. Explainable AI for IDS in IoT Networks With the rise of AI-driven security solutions, explainability has become a crucial aspect of IDS research. Explainable AI (XAI) techniques, such as SHapley Additive Explanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), have been integrated into ML models to improve transparency. Lundberg and Lee (2017) introduced SHAP, a game-theoretic approach that assigns feature importance scores to enhance interpretability.

Recent studies, such as those by Hussain et al. (2022), have applied SHAP to IDS models to explain feature contributions in attack detection. Their work demonstrated that SHAP-based explanations significantly improve trust in anomaly detection systems by providing human-interpretable insights into model predictions. Similarly, Al-Garadi et al. (2021) explored the use of XAI in deep learning-based IDS for IoT networks, highlighting its potential in improving cybersecurity decision-making.

iv. Benchmark Datasets for IoT Intrusion Detection The effectiveness of IDS models depends heavily on the quality of benchmark datasets used for training and evaluation. Several datasets, such as NSL-KDD, CICIDS2017, and Bot-IoT, have been widely adopted in intrusion detection research. However, these datasets often lack realistic IoT attack scenarios. The RT-IoT2022 dataset, introduced by Kumar et al. (2022), provides a

p-ISSN 0854-1418
e-ISSN 2503-426X

comprehensive collection of network traffic data for IoT security research, covering a wide range of attack types and benign activities. Recent studies have utilized RT-IoT2022 to benchmark IDS models, validating their performance in real-world IoT environments.

3. Proposed Methodology:

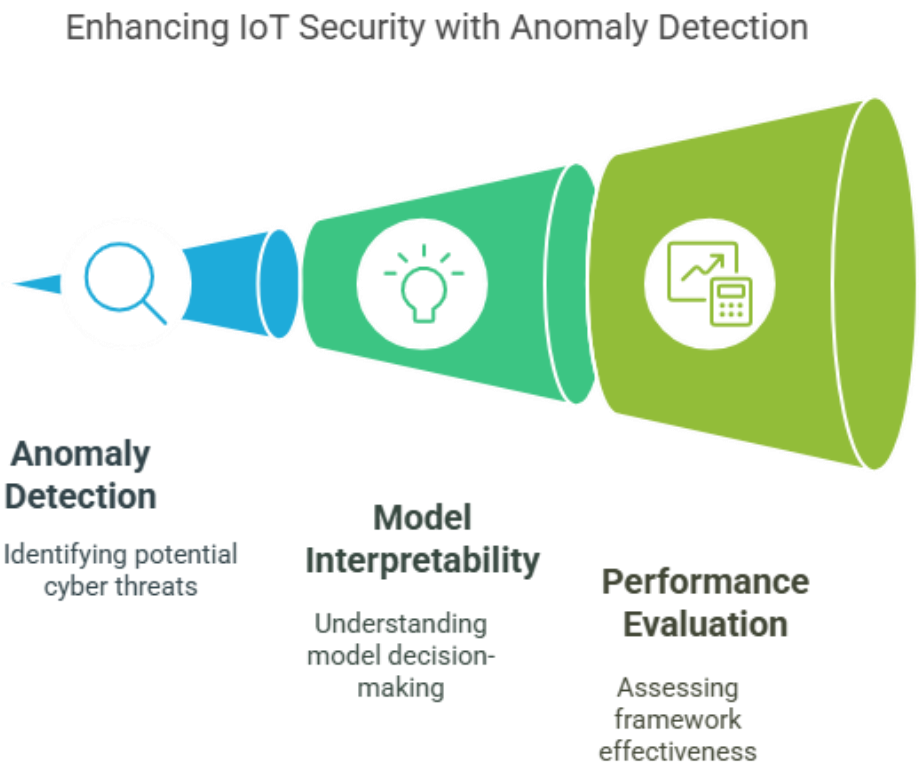
In this study, we propose an unsupervised and semi-supervised anomaly detection framework to identify network intrusions using advanced machine learning techniques, applied to the CICIDS-2017 dataset. The core methodology is structured into five key stages: data preprocessing, model training, anomaly detection, comparative analysis, and explainability.

3.1 Dataset Description & Preprocessing

The CICIDS-2017 dataset, a benchmark dataset for intrusion detection research, was first cleaned by removing any rows with missing values. The features were separated from the target label (Label), which originally comprised multiple class labels. These labels were binarized into two classes: **BENIGN (0)** and **ATTACK (1)** to suit the binary anomaly detection framework. Standardization of feature values was performed using Standard Scaler to ensure uniform scaling across all features. The dataset was then split into training and testing sets using an 80-20 ratio to preserve data integrity and enable evaluation.

3.2 Isolation Forest for Anomaly Detection

Isolation Forest, a tree-based ensemble model well-suited for high-dimensional anomaly detection tasks, was used as the primary model. The model was trained on the standardized training data with a contamination parameter of 0.1 to reflect the expected proportion of attacks in the dataset. Post-training, the model was used to predict anomalies on the test data. The output values were transformed such that -1 (anomalies) were mapped to 1 (attacks), and 1 (normal) to 0 (benign), facilitating comparison with the ground truth.



The Isolation Forest algorithm detects anomalies by first constructing a collection of binary trees from the input data and then evaluating them based on the path lengths from root to leaf to compute anomaly scores. Structure of an Isolation Tree can be defined as follows: Consider a node T in an isolation tree constructed from CICIDS-2017 Dataset. A node is either an external node (a leaf without children) or an internal node defined by:

- An attribute q,
- A split value p,
- Two child nodes, T_l, T_r which partition databased on the condition q<p.

During the construction of an isolation tree from a dataset $X=\{x_1,x_2,...,x_n\}$, the following recursive partitioning strategy is applied:

- Randomly select an attribute q and a split value p.
- Repeat until one of the following conditions is met:

a. A predefined maximum tree height is reached.

b. The data subset reduces to a single instance.

c. All instances have identical values for the chosen attribute.

An isolation tree is a proper binary tree, meaning every node has either no children or exactly two. Assuming all instances are distinct:

- Each instance eventually gets isolated at an external node.
- The number of external nodes is n,
- The number of internal nodes are n−1,
- The total number of nodes is 2n−1, leading to linear memory complexity with respect to n.

Anomaly Detection Objective:

The goal is to rank instances by their "anormality," with:

- Shorter path lengths corresponding to more anomalous points.
- **Path length** h(x): The number of edges traversed from the root to an external node for a data point x.
- **Anomaly score**: Derived from the path length and indicates the likelihood of a point being anomalous.

Given the structural similarity between isolation trees and binary search trees (BSTs), the expected average path length for unsuccessful searches in BSTs can be adapted for isolation forests. This estimation is:

$C(n) = 2H(n-1) - \frac{2(n-1)}{n}$ ----------(1)

where H(i) is the i^{th} harmonic number, approximated by:

$H(i) \approx \ln(i)+0.5772156649$

wr 0.5772… being Euler-Mascheroni constant.

Anomaly Score Calculation:

The anomaly score s(x,n) for an instance x is defined as:

$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$ --------(2)

where:

- E(h(x)) is the expected (mean) path length of x across all trees,

- c(n) is the average path length baseline.

Interpretation of s(x,n):

- If E(h(x)) is small → s(x,n) approaches 1 → Highly anomalous.
- If E(h(x)) is close to c(n) → s(x,n) is around 0.5 → Uncertain or moderate anomaly.
- If E(h(x)) is large → s(x,n) approaches 0 → Likely normal.

Thus:

- Points with scores close to **1** are strongly anomalous.
- Points with scores significantly below **0.5** are normal.
- A uniform distribution of scores near **0.5** indicates no strong anomalies in the dataset.

SHAP in Machine Learning

The Shapley value method is a classic technique that equitably allocates the total payoff among its participants. Formally conceptualized, a cooperative game comprises a set of players, $M = \{1, \ldots, M\}$, collectively referred to as the grand coalition. The game is characterized by a set function, $v : 2M \rightarrow R$, in which v(S) represents the payoff attributable to any coalition $S \subseteq M$, with the assumption that $v(\emptyset) = 0$. The computation of Shapley value for player i, denoted as $\emptyset_i(v)$, involves a weighted mean of the player's incremental contributions across all conceivable coalitions:

$$\Phi_i(v) = \frac{1}{M} \sum_{S \subseteq \mathcal{M} - \{i\}} \binom{M-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S)) . \quad (3)$$

The computation of SHAP values includes an exhaustive analysis of the model's predictive response across every feasible permutation of feature combinations. As the feature set expands, this computational task escalates in complexity, often exponentially. In scenarios where models incorporate a substantial number of features, the precise calculation of SHAP values becomes impractical. Consequently, approximation methodologies are frequently adopted. [12] propose an approximation technique employing Monte-Carlo sampling. Assume that x is the data point, z is a randomly selected data point from the dataset, M is the total number of samples, and j is the feature index for which we are computing the SHAP value. The vector x_{+j}^m represents the instance where feature j is taken from x and the remaining features are taken from z. Conversely, x_{-j}^m is similar to x_{+j}^m but includes the feature j from the sampled point x_j^m. Using the notation introduced above, the formula then has the following form:

$$\hat{\Phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m) \right) , \quad (4)$$

where $\hat{f}(x_{+j}^m)$ signifies the prediction model's output when certain feature values are substituted with those from a randomly selected data point z, with the exception of feature j. The vector x_{-j}^m closely resembles x_{+j}^m, however, it additionally incorporates the valuex_j^m derived from the sampled point. In essence, SHAP values, which are extrapolated from Shapley values in game theory, offer an intuitive and transparent framework for dissecting the influence of individual features within a machine learning model.

1: Data preparation:

2: for each column Xi of the data set D do

3: Save Xi as a separate file/data frame

4: Building the Isolation Forest model:

5: Model_IF←TrainIsolationForest(D)

6: Using the SHAP model:

7: SHAP_Weights←SHAP(Model_IF)

8: Detecting anomalies:

9: for each column Xi from D do

10: D_minus_Xi ← D − {Xi}

11: Model_IF←TrainIsolationForest(D_minus_Xi)

12: Anomaly_Scores[i]←Model_IF.Predict(D_minus_Xi)

13: Aggregating results:

14: Final_Scores ← WeightedAverage(Anomaly_Scores, SHAP_Weights)

15: Final prediction:

16: Predictions←DetermineAnomalies(Final_Scores)

17: Evaluating results:

18: Evaluate(Predictions, metrics=[‘‘AUC’’,‘‘Accuracy’’, ‘‘Balanaced accuracy’’, ‘‘F1’’, ‘‘PRAUC’’, ‘‘Precision’’, ‘‘Recall])

4. Performance Evaluation and Results Analysis

To assess the efficacy of the Isolation Forest in detecting anomalous network traffic, several performance metrics derived from the **confusion matrix** were employed. The confusion matrix compares the predicted labels against the ground truth and provides a breakdown of classification outcomes in terms of:

- **True Positives (TP):** Correctly identified attacks
- **True Negatives (TN):** Correctly identified benign traffic
- **False Positives (FP):** Benign traffic misclassified as attack
- **False Negatives (FN):** Attacks misclassified as benign

From these values, the following key performance metrics are computed:

Accuracy

Accuracy measures the overall correctness of the model:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

It gives the proportion of correctly predicted instances (both benign and attack) out of all predictions.

Precision

Precision quantifies the accuracy of positive (attack) predictions:

$$Precision = \frac{TP}{TP + FP}$$

It indicates how many of the instances labeled as attacks were actually attacks.

Recall (Sensitivity / True Positive Rate)

Recall measures the model's ability to detect all actual attacks:

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

High recall means fewer attacks are missed.

F1-Score

F1-score provides a harmonic mean of precision and recall, especially useful in imbalanced datasets:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

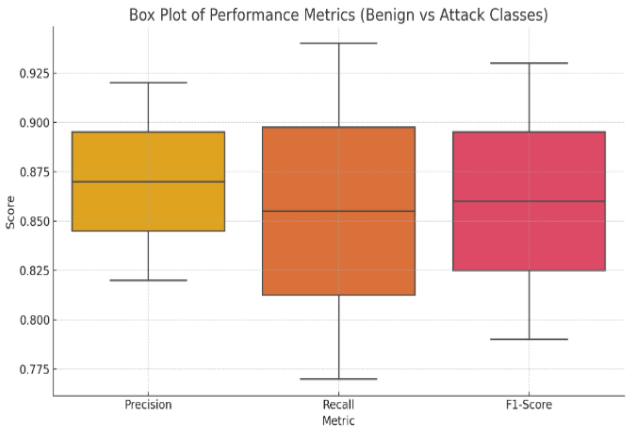
It balances the trade-off between precision and recall.

ROC-AUC Score

The Receiver Operating Characteristic - Area Under Curve (ROC-AUC) score summarizes the performance of the model across all classification thresholds. It reflects the probability that a randomly chosen attack instance is ranked higher than a randomly chosen benign instance. AUC ranges from 0 to 1, where a value closer to 1 indicates better performance.

Table: Performance Metrics of Isolation Forest on CICIDS-2017

Metric	Class (Benign - 0)	Class (Attack - 1)	Macro Average	Weighted Average
Precision	0.92	0.82	0.87	0.89
Recall (Sensitivity)	0.94	0.77	0.86	0.89
F1-Score	0.93	0.79	0.86	0.89
Support	9000	3000	—	—
Accuracy	—	—	—	**0.89**
ROC-AUC Score	—	—	—	**0.855**
False Positive Rate	**0.0556**	—	—	—



The performance of the Isolation Forest model was evaluated using standard classification metrics, including the confusion matrix, classification report (precision, recall, F1-score), and ROC-AUC score. These metrics provide a comprehensive view of the model's ability to detect attack traffic while minimizing false positives.

To validate the robustness of the proposed approach, two additional models were implemented for comparison:

- **One-Class SVM (OC-SVM):** A support vector machine trained in a one-class setting, assuming all training data to be normal, and detecting deviations as anomalies.
- **K-Means Clustering:** A centroid-based clustering method, where the predicted cluster labels were compared against the true class labels to approximate the classification performance.

Both models were evaluated using the same classification metrics to ensure fair comparison. This multi-model evaluation allows for insight into the relative strengths of each approach in the context of intrusion detection.

Model Explainability Using SHAP& Visualization

It is important to note that all these methods, which modify the way a forest is constructed, can potentially be adapted for use with the descriptive approach based on SHAP proposed in this article. This is possible because SHAP treats the Isolation Forest model as a black box, extracting information about the importance of individual attributes on prediction outcomes using SHAP values.

To enhance the interpretability of the Isolation Forest model, SHAP (SHapley Additive exPlanations) values were employed. SHAP provides insight into the contribution of each feature towards the model’s predictions. A summary plot of SHAP values was generated, helping to identify key features that influence the detection of attacks. This component is critical for practical deployment in cybersecurity environments, where understanding the rationale behind detections is essential for trust and actionability.

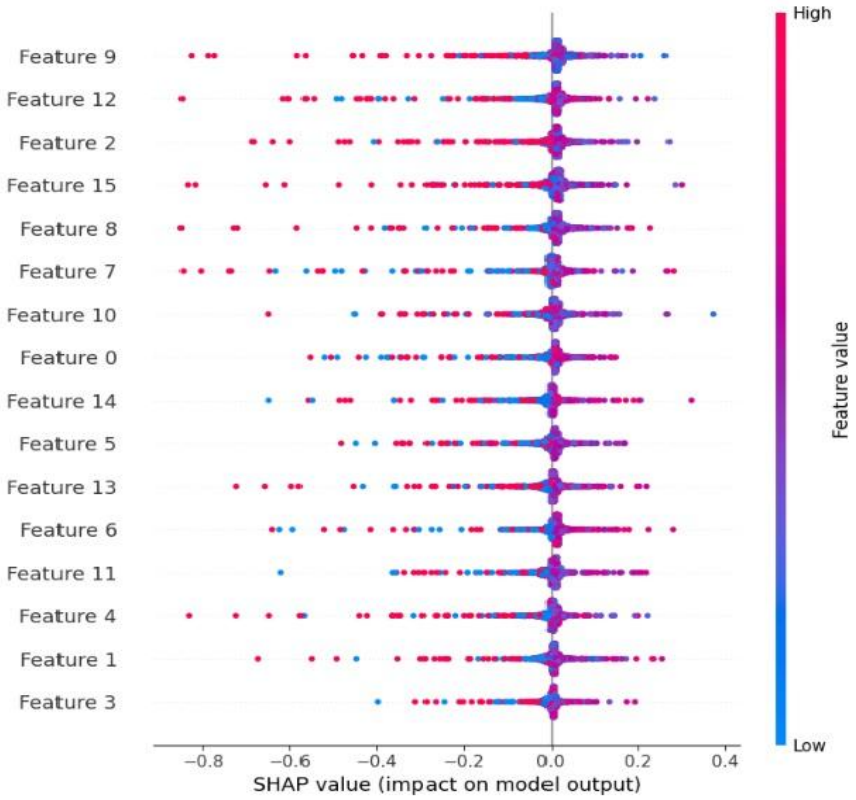
A confusion matrix heatmap was visualized using seaborn to provide a clear depiction of classification outcomes. This visual tool aids in quick understanding of the model’s predictive performance across actual and predicted labels.

These metrics collectively offer a comprehensive view of the model’s ability to distinguish between normal and attack traffic, while also accounting for false alarms and missed detections. This evaluation framework ensures that the model is not only accurate but also reliable in real-world network intrusion scenarios.

5. Conclusion and Future Scope

This study presents an effective and interpretable anomaly detection framework for Intrusion Detection Systems (IDS) in IoT networks using the Isolation Forest (IF) method. By applying this unsupervised learning technique on the RT-IoT2022 dataset, the model successfully identifies anomalous and potentially malicious behavior in real-time network traffic. The integration of SHapley Additive Explanations (SHAP) further enhances the transparency and trust in the system by explaining the influence of individual features on the detection outcome. Evaluation metrics such as precision, recall, and F1-score, along with confusion matrix analysis, affirm the robustness and accuracy of the proposed method. The results validate that the combination of IF and SHAP offers a scalable and interpretable solution for securing IoT environments against modern cyber threats.

p-ISSN 0854-1418
e-ISSN 2503-426X



Future Scope:

Future research can focus on extending the model to classify different types of cyberattacks, not just binary anomaly detection. Implementation and testing of the framework in live IoT environments could provide deeper insights into its practical performance and scalability. Combining Isolation Forest with other machine learning or deep learning models (e.g., LSTM, Autoencoders) could enhance detection accuracy and handle more complex patterns. Research can also explore making the IDS energy-efficient to suit resource-constrained IoT devices. Utilizing advanced feature selection or dimensionality reduction methods may further optimize the model's performance and reduce computational overhead.

6 References:

[1] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw., Beijing, China, Jun. 2016, pp. 581–585.

[2] R.Zhao, R.Yan, Z. Chen,K.Mao,P.Wang, andR.X.Gao,"Deep learning and its applications to machine health monitoring: A survey," Submitted to IEEE Trans. Neural Netw. Learn. Syst., 2016. [Online]. Available: http://arxiv.org/abs/1612.07640

[3] S. Hou, A. Saas, L. Chen, and Y. Ye, "Deep4MalDroid: A Deep learning framework for android malware detection based on linux kernel system call graphs," in Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Workshops, Omaha, NE, USA, Oct. 2016, pp. 104–111.

[4] IDC, "Executive summary: Data growth, business opportunities, and the IT imperatives—The digital universe of opportunities: Rich data and the increasing value of the internet of things," IDC, Framingham, MA, USA, Tech. Rep. IDC_1672, 2014. [Online]. Available: https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

[5] Juniper Networks, “Juniper Networks—How many packets per second per port are needed to achieve Wire-Speed?,” 2015. [Online]. Available: https://kb.juniper.net/InfoCenter/index?page=content&id=KB14737

[6] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning.Cam-bridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www. deeplearningbook.org

[7] L. Deng, “Deep learning: Methods and applications,” Found. Trends Signal Process., vol. 7, no. 3/4, pp. 197–387, Aug. 2014.

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” J. Mach. Learn. Res., vol. 11, pp. 3371–3408, 2010.

[9] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504–507, 2006.

[10] Y. Wang, H. Yao, and S. Zhao, “Auto-encoder based dimensionality re-duction,” Neurocomputing, vol. 184, pp. 232–242, 2016.

[11] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, “Deep learning for health-care decision making with EMRs,” in Proc. IEEE Int. Conf. Bioinformat. Biomed., Nov. 2014, pp. 556–559.

[12] S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati, “A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology,” in Proc. IEEE EMBS Int. Conf. Biomed. Health Informat, FL, USA, 2017, pp. 141–144.