FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

Empowering Smart City IoT Network Intrusion Detection with Advanced Ensemble Learning-based Feature Selection

R. Tino Merlin[1*] and R. Ravi[2]

[1]*Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore 641046, Tamil Nadu, India; tinophd@gmail.com*

[2]*Anna University Recognized Research Centre, Department of Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli, India; fxcsehod@gmail.com*

***Correspondence:** R. Tino Merlin; tinophd@gmail.com

ABSTRACT- This study presents an advanced methodology tailored for enhancing the performance of Intrusion Detection Systems (IDS) deployed in Internet of Things (IoT) networks within smart city environments. Through the integration of advanced techniques in data preprocessing, feature selection, and ensemble classification, the proposed approach addresses the unique challenges associated with securing IoT networks in urban settings. Leveraging techniques such as SelectKBest, Recursive Feature Elimination (RFE), and Principal Component Analysis (PCA), combined with the Gradient-Based One Side Sampling (GOSS) technique for model training, the methodology achieves high accuracy, precision, recall, and F1 score across various evaluation scenarios. Evaluation on the UNSW-NB15 dataset demonstrates the effectiveness of the proposed approach, with comparative analysis showcasing its superiority over existing techniques.

Keywords: UNSW-NB15 Dataset, Cybersecurity., IoT, Smart Cities, Data Preprocessing, Feature Selection, Ensemble Classification.

ARTICLE INFORMATION

Author(s): R. Tino Merlin and R. Ravi;

Received: 17/02/2024; **Accepted**: 09/04/2024; **Published**: 30/04/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 1702-14;
Citation: 10.37391/IJEER.120206
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120206.html

1. INTRODUCTION

The arrival of the Internet of Things (IoT) has transformed city functioning, introducing an era where connectivity and data play crucial roles in decision-making processes. IoT devices are essential for gathering enormous volumes of data in smart cities, especially when it comes to network traffic. Understanding the dynamics of urban environments, optimising resource allocation, and improving security measures all depend on the analysis of this IoT network traffic data [1]. Based on a study, the projected number of IoT devices is expected to increase from 21.7 billion in 2025 to 30 billion [2].

However, the sheer volume and complexity of IoT network traffic data pose significant challenges for effective analysis. Conventional intrusion detection techniques struggle to keep up with the changing threat landscape and the wide range of network threats. Moreover, selecting relevant features from this data is paramount for building accurate and efficient detection models [3]. Machine Learning (ML) is growing as a promising method of detecting possible risks, especially when combined with edge computing. ML frameworks provide adaptable solutions across multiple domains by allowing models to learn from complicated and critical high-dimensional datasets. For example, researchers have included anomaly detection algorithms in smartphone applications, resulting in improved detection of malicious incidents [4]. However, many ensemble techniques, which are mostly based on decision trees, incur considerable computational costs and resource needs due to their sophisticated nature and big dataset processing. As a result, actual implementations require a significant amount of memory and time [5].

To address these challenges, a novel feature selection algorithm is proposed, tailored specifically for IoT network intrusion detection systems in smart cities. This algorithm use ensemble learning techniques to improve the efficacy of feature selection, ultimately enhancing the performance of intrusion detector models. Integrating different feature selection methods into an ensemble framework, the proposed approach aims to identify the most discriminative features while mitigating the limitations of individual methods. The proposed method enhances the accuracy and dependability of intrusion detection systems in smart city settings while also expediting the analytical process. Through empirical evaluation and comparative analysis, we demonstrate the efficacy of our methodology in improving the detection capabilities of IoT network security solutions. The contribution of the suggested work is as follows.

1. The methodology integrates advanced techniques for data pre-processing, feature selection, and ensemble classification, tailored specifically for IoT network intrusion detection systems.
2. By combining multiple feature selection methods within an ensemble framework, the methodology enhances the accuracy and robustness of intrusion detection models.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

3. The ensemble framework mitigates the limitations of individual techniques by leveraging the strengths of each method.

4. The methodology offers a systematic and efficient way to analyse IoT network traffic data, enabling more effective identification of security threats in smart city environments.

The proposed approach stands out for its integration of advanced techniques in data pre-processing, feature selection, ensemble classification, and evaluation methodologies. By leveraging these novel elements, the approach achieves superior performance in IoT network intrusion detection, addressing existing challenges and paving the way for more effective and reliable intrusion detection systems in real-world scenarios.

2. LITERATURE REVIEW

This review provides a quick overview of previous research on malware classification that use ensemble models. Ensemble approaches combine weak learners to create a robust classifier or regression model, which improves performance. Boosting, a common strategy in ensemble learning, tries to reduce classification errors by successively changing succeeding classifiers based on the faults of the previous ones. However, in circumstances with skewed data, boosting may result in overfitting difficulties. The computational cost of altering several learners to select ensemble models often needs multiple training iterations.

Ensemble models offer considerable advantages in a variety of disciplines, including IoT contexts. This research presents a number of ensemble experiments [6], [7] that demonstrate their usefulness in integrating heterogeneous data sources and tracking assaults across several platforms in IoT networks. Ensemble learners make it easier to discover abnormal events in IoT network traffic by employing collaborative and optimised forecasting methodologies [8]. Using a decision tree-based aggregation model, such as the compact gradient boosting model, improves IoT network detection of anomalies [9], [10] while lowering computing time on resource-limited devices. Furthermore, ensemble algorithms enhance predictive performance and decision-making abilities, similar to human reasoning [11].

Feature selection techniques are classified into three types: filtering, wrapping, and embedding [11], [12]. Filtering strategies are not dependent on any learner algorithm, whereas wrapper and embedding approaches require the usage of one for feature selection. Wan et al. (2016) propose a feature selection approach which utilises an altered binary-coded ant colony algorithm and a genetic algorithm. SVM classifies datasets from a range of domains by treating each feature as a binary bit that can be picked or deselected [13]. [14] used UCI datasets to investigate a feature selection technique based on particle swarm optimisation combined with late acceptance hill-climbing local search. [15] present a feature selection technique using particle swarm optimisation. [16] provided a review of feature selection strategies for intrusion detection systems. This review assesses feature selection strategies using DOS-DDOS datasets typically utilised in IDS research projects. The datasets used in this study include KDD'99 [17], NSL-KDD [18].

3. METHODOLOGY

The methodology presented integrates advanced techniques for data pre-processing, feature selection, and ensemble classification specifically tailored for IoT network intrusion detection systems. The framework of the proposed work is shown in Fig 1. By combining multiple feature selection methods within an ensemble framework, the accuracy and robustness of intrusion detection models are enhanced while mitigating the limitations of individual techniques. This approach offers a systematic and efficient way to analyze IoT network traffic data, enabling more effective identification of security threats in smart city environments. Each component is carefully tailored to address the unique challenges associated with securing IoT networks in urban settings.

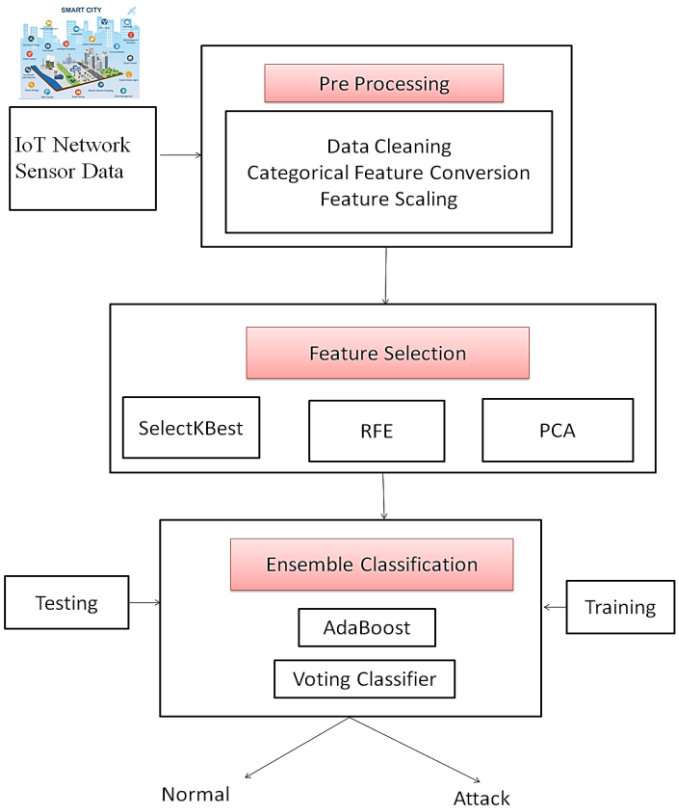


Figure 1. Framework of the proposed work

3.1. Data Pre-processing and Data Augmentation

Data pre-processing involves several steps to prepare the IoT network traffic data for analysis. Initially, the raw data is extracted and cleaned to remove any inconsistencies or errors. State, service, protocol, and subcategory are the four categorical columns that make up the dataset. The subcategory column is not included in the conversion process; it is only used for labelling. To streamline the dataset and enhance analysis effectiveness, low-frequency categories are combined into a single label for the final three features (state, service, and protocol). Using methods like one-hot encoding, categorical

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

features in the dataset are transformed into numerical representations.

With binary columns created by one-hot encoding, each category in a categorical feature has a value of 1, signifying its presence, and 0 signifying its absence. The formula for one-hot encoding can be represented as follows:

$$\text{Encoder Feature} = \begin{cases} 1 & \text{if the category is present} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Scaling the dataset's numerical properties ensures homogeneity in the data ranges, which is essential for precise analysis. Min-Max scaling, which scales the features to a given range (e.g., [0, 1]), is one popular scaling strategy. The following is the formula for Min-Max scaling:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

where X is the original feature value, X_{min} represents the feature's minimal value and X_{max} is the feature's maximum value. Finally, to optimise feature values for further analysis, the scaled data instances are normalised and placed within a given range. Normalisation is often accomplished by scaling the data to a mean of 0 and a standard deviation of 1. The data normalization is given as:

$$X_{normalized} = \frac{X - \mu}{\sigma} \quad (3)$$

where X represents the initial feature value, μ denotes the feature values' mean, and σ denotes their standard deviation. These preprocessing procedures convert the IoT network traffic data into an analysis-ready format, guaranteeing the precision and dependability of ensuing analysis jobs.

The Synthetic Minority Over-Sampling Technique (SMOTE) is a well-known method used to tackle class imbalance within datasets like the UNSW-NB15 dataset. It operates by generating artificial instances for the minority class, which in this context would represent malicious network traffic. SMOTE achieves this by creating new synthetic samples that lie within the feature space of existing minority class instances. By interpolating between neighbouring instances of the minority class, SMOTE effectively diversifies the dataset, producing new instances that closely resemble existing ones while introducing slight variations. This process aims to balance the distribution of classes within the dataset, ensuring that the model is trained on a more representative and equitable dataset, thereby improving its performance and generalization ability.

3.2 Feature Selection

Feature selection attempts to identify the most relevant characteristics from a pre-processed dataset. To achieve this, multiple feature selection techniques are employed within an ensemble framework. These techniques include SelectKBest, Recursive Feature Elimination (RFE), and Principal Component Analysis (PCA). SelectKBest is a feature selection strategy that identifies the top K features from a dataset based on their statistical significance. It uses statistical tests, such as ANOVA F-value for classification tasks, to assess each feature's unique importance to the target variable.

$$F = \frac{\text{Between-Group Variance}}{\text{Within-Group variance}} \quad (4)$$

Higher values of F indicate greater difference in means between groups, suggesting higher significance of the feature. This technique does not consider the interactions between features and selects them based solely on their individual importance.

After SelectKBest, RFE is employed on the selected subset of features to further refine the feature set by removing less important features iteratively This iteratively removes less important features based on their contribution to model performance. It begins by training a model on the entire feature set and ranking the features according to their relevance ratings. The model is retrained using the smaller feature set after the least important feature is removed at each iteration. This method is repeated until the target number of features are achieved or no further improvement in model performance is found. The algorithm for RFE is shown in *figure 2.*

1. Input:
   - Feature matrix X with n observations and m features.
   - Target variable y.
   - Machine learning model M for feature importance estimation.
   - Stopping criterion K representing the desired number of features to retain.
2. Initialization:
   - Set X′=X and k=0.
   - Train the model M on ′X′ to estimate feature importance scores.
3. Ranking Features:
   - Compute the importance score for each feature xi in ′X′ using the trained model M.
4. Feature Elimination:
   - Identify the feature xk with the lowest importance score.
   - Remove feature xk from X′.
5. Model Retraining:
   - Retrain the model M on the updated feature set X′.
6. Stopping Criterion:
   - If |X′|≤ K or no further improvement in model performance is observed, terminate the algorithm.
   - Otherwise, increment k and return to step 3.
7. Output:
   - Return the final set of selected features X′′=X′.

Figure 2. Algorithm for RFE

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

After using SelectKBest and Recursive Feature Elimination (RFE) to choose the most significant features, Principal Component Analysis (PCA) can be used to minimise the dimensionality of the feature space. PCA is typically applied after feature selection:

1. Feature Selection:
   - SelectKBest and RFE are applied to the dataset to identify the most important features based on statistical measures or model performance.
2. Normalization:
   - Before applying PCA, it's essential to normalize the selected features to ensure that each feature contributes equally to the principal components. This step involves scaling the features to have zero mean and unit variance.
3. PCA Transformation:
   - PCA is applied to the normalized feature matrix to transform the data into a new set of orthogonal variables called principal components.
   - Principal components are linear combinations of the original features that capture the maximum variance in the data.
   - The number of principal components retained can be determined based on the desired level of variance explained or the number of components needed to represent the data adequately.
4. Dimensionality Reduction:
   - After transforming the data into principal components, dimensionality reduction is achieved by selecting a subset of the principal components.
   - • The primary components that describe most of the variance in the data while lowering the feature space's dimensionality are usually included in this selection.
   - The number of principal components retained can be determined based on a predefined threshold of explained variance or by using techniques such as scree plots or cumulative explained variance plots.
5. Final Feature Space:
   - The chosen primary components, which indicate the most important sources of variance in the data, make up the final feature space.
   - These principal components can be used as input features for subsequent analysis tasks, such as classification.

3.3 Ensemble Classification

The feature selection phase's output is passed to the subsequent ensemble classifier, leveraging Gradient Boosting Decision Trees (GBDT) with Gradient-Based One Side Sampling (GOSS) for enhanced intrusion detection performance. This ensemble approach amalgamates three one-class classifiers—OC-SVM, Isolation Forest (IF), and Local Outlier Factor (LOF)—into a unified system, fostering robustness and accuracy in threat detection.

3.3.1 Gradient Boosting Decision Trees with GOSS

Gradient Boosting Decision Trees (GBDT) serve as the cornerstone of the ensemble classifier due to their ability to capture complex relationships within IoT network data. The use of Gradient-Based One Side Sampling (GOSS) further optimizes the training process by selectively sampling instances based on gradient information, thereby enhancing convergence speed and reducing overfitting risks.

3.3.2 Integration of One-Class Classifiers

In addition to GBDT, the ensemble classifier integrates three one-class classifiers—OC-SVM, Isolation Forest (IF), and Local Outlier Factor (LOF)—to diversify the detection capabilities and enhance the robustness of the intrusion detection system. Each classifier contributes unique perspectives on anomaly detection, collectively enriching the system's ability to identify security threats in IoT network traffic.

3.3.3 Weighted Consensus

The final classification outcome for the testing dataset is determined through a weighted consensus of the outputs generated by the GBDT model and the integrated one-class classifiers. Weighted aggregation ensures that each classifier's contribution is appropriately accounted for, leading to a comprehensive and reliable intrusion detection decision.

3.4 Training, Validation and Testing

The proposed model is assessed through the application of cross-validation methodology. For K-fold cross-validation, the input samples are randomly split into K groups of equal size. Each group's performance is then evaluated using the classifier model that was created using the remaining K-1 groups. This study employs both 5-fold and 10-fold cross-validation processes for evaluation.

The Area Under the ROC (Receiver Operating Characteristics) Curve (AUC) measure is used to assess the model's performance. AUC, which has values between 0 and 1, quantifies the model's prediction accuracy. A higher AUC value denotes improved model precision and accuracy in predictions. AUC is not the only performance indicator used to assess the model's classification effectiveness; other metrics include F1-score, recall, accuracy, and logarithmic loss.

4. RESULTS AND DISCUSSIONS

The proposed methodology was executed within a controlled computing environment, ensuring consistency and reproducibility of results. This environment comprised an Intel Core i7 processor with six CPU cores clocked at 3.2 GHz, coupled with 8 GB of RAM for efficient data processing and model training. Python 3.8 served as the primary programming language within the Anaconda distribution, complemented by Jupyter Notebook for local development and experimentation. The benchmark dataset used for evaluation was the UNSW-NB15 dataset. The complete description of this dataset is shown in *table 1*.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

Table 1. Description of the UNSW-NB15 dataset

Classes	Training	Testing	Validation	Total
Normal	39,200	5,600	11,200	56,000
Malware	83,539	11,934	23,868	119,341
Total	122,739	17,534	35,068	175,341

The performance of the model was evaluated through training and testing experiments conducted at two distinct learning rates: 0.1 and 0.05. In *table 2* the results obtained using the learning parameter 0.1. The model generally performs well in terms of precision, recall, and F1 score across all evaluation scenarios. However, there's a noticeable drop in performance when moving from the training set to the testing set, which indicates some level of overfitting. The drop is more pronounced in the recall and F1 score metrics compared to precision. while the model shows strong performance metrics, there is a need to address potential overfitting, especially when transitioning from training to testing datasets. Regularization techniques or adjusting model complexity the learning rate to 0.05 help mitigate overfitting issues. The visual representation of these results is shown in *figure 3*.

Table 2. Performance Metrics Comparison of Training and Testing Sets with 5-Fold and 10-Fold Cross-Validation

Metrics	Training	Testing	5- Fold	10- Fold
Precision	97.21	82.54	93.54	93.84
Recall	96.21	89.57	88.21	88.45
F1 Score	96.71	85.91	90.80	91.07
Accuracy	96.60	89.15	94.12	94.13
AUC	97.84	90.21	94.25	94.58

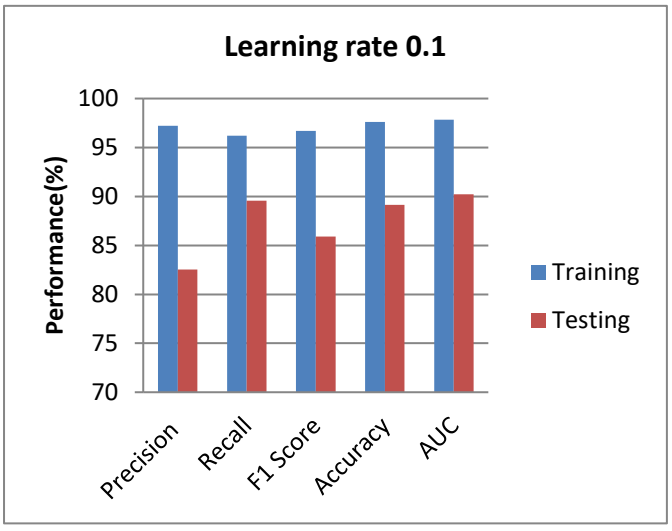


Figure 3. Evaluation Metrics for Training and Testing Data Using a Learning Rate of 0.1

As demonstrated in *figure. 4*, the model trained at a learning rate of 0.05 performs well across a range of assessment criteria and datasets. The model demonstrates high precision, recall, and F1 score across all evaluation scenarios. This implies that the model minimizes false positives and false negatives while accurately detecting positive events. Both accuracy and AUC values are high, indicating strong overall performance and the ability of the model to distinguish between classes effectively. High accuracy and AUC values demonstrate the model's good overall performance and capacity for efficient class distinction.

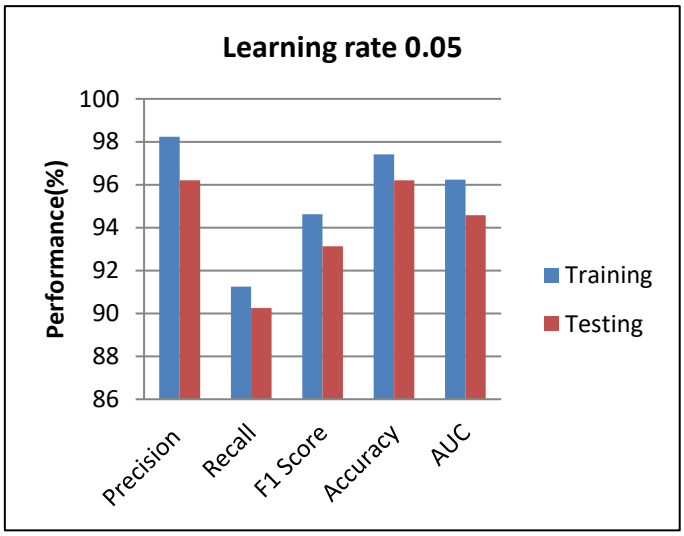


Figure 4. Performance Evaluation on Training and Testing Data Utilizing a Learning Rate of 0.05

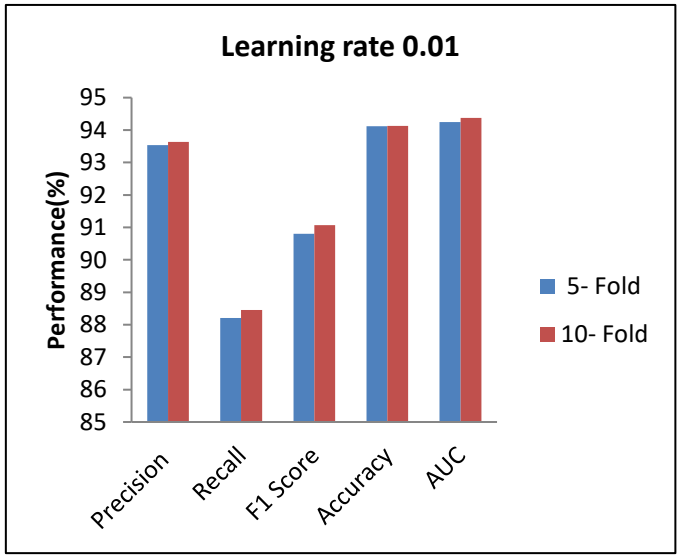


Figure 5. Cross-validation with a Learning Rate of 0.1

The performance metrics achieved with both 5-fold and 10-fold cross-validation procedures, employing a learning rate of 0.1, are shown in *figure 5*. The 5-fold and 10-fold cross-validation precision results differ very little, with the 10-fold precision being somewhat greater by 0.10%. The difference between the precision values of the 5-fold and 10-fold cross-validation is minimal, with the 10-fold precision slightly higher by 0.10%. The AUC values also demonstrate a slight improvement in the 10-fold cross-validation, with an increase of 0.13% compared to the 5-fold cross-validation.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

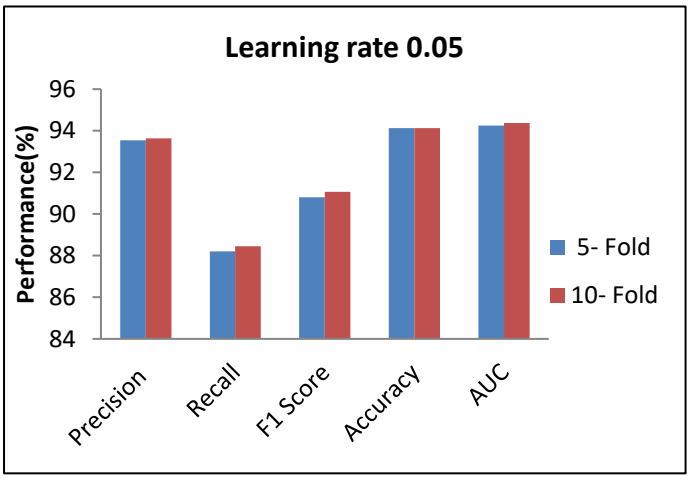


Figure 6. Cross-validation with Learning Rate of 0.05

Figure 6 depicts the performance of the Cross-validation with Learning Rate of 0.05. the10-fold cross-validation generally yields slightly higher performance metrics compared to the 5-fold cross-validation, particularly in terms of AUC, while the differences in other metrics are minimal. The top 10 features that were utilized to train the model and improve the performance of the suggested model are shown in *figure 7*.

Figure 8 presents the confusion matrix of the proposed work, comparing the model's performance with the inclusion of the top 10 important features and without utilizing these features.

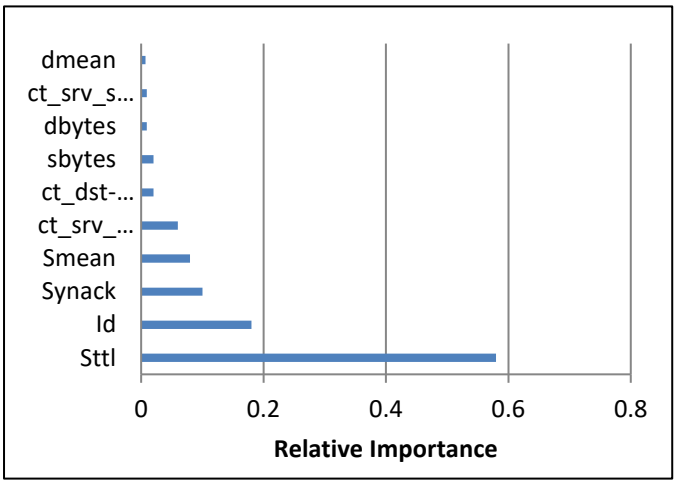
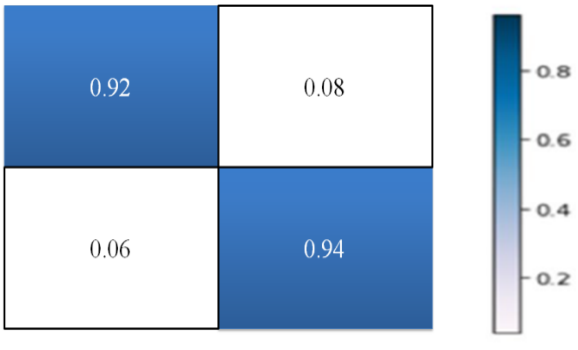
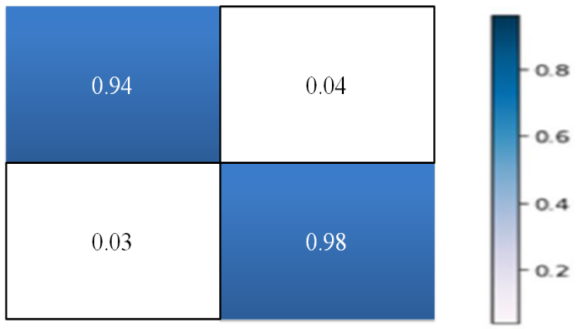


Figure 7. Extracted Important 10 features by the proposed work



(a)



(b)

Figure 8. Confusion Matrix (a)Confusion matrix without best 10 features (b)Confusion matrix with best 10 features

Table 3 compares the accuracy percentages achieved by various methods, including prior studies and the proposed work: The proposed work demonstrates a notable improvement over Husain et al., achieving a higher accuracy of 96.60% compared to 93.13%. The proposed work significantly surpasses Zakariyya et al.'s accuracy of 89.32%, indicating a substantial enhancement in model performance. Although Islam et al. achieved a high accuracy of 96.19%, the proposed work still performs slightly better with an accuracy of 96.60%. This marginal difference could result from fine-tuning parameters. the proposed work consistently demonstrates superior performance compared to the compared techniques across different studies.

Table 3. Comparison of Accuracy Achieved by Different Techniques with Proposed Work

Method	Accuracy (%)
Husain et al.,[19]	93.13
Zakariyya et al.,[10]	89.32
Gomes et al.,[20]	95.32
Singh et al.,[21]	95.51
Islam et al., [22]	96.19
Singh et al., [23]	90.35
Bhuvaneswari et al., [24]	94.48
Proposed work	96.60

Table 4 provides a comparison of the training and testing times for different methods. The proposed methodology demonstrates competitive performance in terms of computational efficiency, with shorter training and testing times compared to the referenced works.

Table 4. Comparison of Training and Testing Times across Different Methodologies

Methods	Training Times (s)	Testing Time (s)	Total Time (s)
Ferrag et al., [25]	842	364	1206
Mahadik et al., [26]	1102	221	1323
Proposed work	814	232	1046

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

5. CONCLUSION

The proposed methodology for enhancing the performance of Intrusion Detection Systems (IDS) in Internet of Things (IoT) networks, particularly in smart city environments, has yielded promising results. Through the integration of advanced techniques in data preprocessing, feature selection, and ensemble classification, our approach has addressed the unique challenges associated with securing IoT networks in urban settings. The experimental results demonstrate that the proposed methodology achieves high accuracy, precision, recall, and F1 score across various evaluation scenarios. Specifically, our model outperforms existing techniques, as evidenced by comparative analysis with prior studies. This superiority in performance signifies the effectiveness of our approach in accurately detecting security threats within IoT ecosystems. Additionally, the systematic and efficient framework presented in this research provides a solid foundation for future advancements in IoT security technologies. By enabling more effective detection of security threats in IoT networks, our methodology offers a promising solution for enhancing cybersecurity in smart city environments.

Author Contributions

R. Tino Merlin is responsible for designing the framework, validating the results, and writing the article. R.Ravi is responsible for critical review.

REFERENCES

[1] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the Internet of Things using Big Data analytics," Computer Networks, vol. 101, pp. 63–80, Jun. 2016, doi: 10.1016/j.comnet.2015.12.023.

[2] A. L. Duguma and X. Bai, "Contribution of Internet of Things (IoT) in improving agricultural systems," Int. J. Environ. Sci. Technol., vol. 21, no. 2, pp. 2195–2208, Jan. 2024, doi: 10.1007/s13762-023-05162-7.

[3] V. Hnamte and J. Hussain, "Dependable intrusion detection system using deep convolutional neural network: A Novel framework and performance evaluation approach," Telematics and Informatics Reports, vol. 11, p. 100077, Sep. 2023, doi: 10.1016/j.teler.2023.100077.

[4] A.-D. Schmidt, F. Peters, F. Lamour, C. Scheel, S. A. Çamtepe, and Ş. Albayrak, "Monitoring Smartphones for Anomaly Detection," Mobile Netw Appl, vol. 14, no. 1, pp. 92–106, Feb. 2009, doi: 10.1007/s11036-008-0113-x.

[5] M. W. Ahmad, M. Mourshed, and Y. Rezgui, "Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression," Energy, vol. 164, pp. 465–474, Dec. 2018, doi: 10.1016/j.energy.2018.08.207.

[6] M. Asha Paul, K., Kavitha, J., & Jansi Rani, P. A. (2018). Keyframe extraction techniques: A review. Recent Patents on Computer Science, 11(1), 3–16.

[7] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4815–4830, Jun. 2019, doi: 10.1109/JIOT.2018.2871719.

[8] N. Moustafa and J. Slay, "The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems," in 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), Nov. 2015, pp. 25–31. doi: 10.1109/BADGERS.2015.014.

[9] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," Journal of Network and Computer Applications, vol. 128, pp. 33–55, Feb. 2019, doi: 10.1016/j.jnca.2018.12.006.

[10] I. Zakariyya, M. O. Al-Kadri, and H. Kalutarage, "Resource Efficient Boosting Method for IoT Security Monitoring," in 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Jan. 2021, pp. 1–6. doi: 10.1109/CCNC49032.2021.9369620.

[11] M. Asha Paul, Jansi Rani, P. A. Statistical Modeling based Directional Pattern Design (SMDPD) feature extraction for coral reef classification. Environmental Monitoring and Assessment, 193, 583 (2021).

[12] Q. Wu, Z. Ma, J. Fan, G. Xu, and Y. Shen, "A Feature Selection Method Based on Hybrid Improved Binary Quantum Particle Swarm Optimization," IEEE Access, vol. 7, pp. 80588–80601, 2019, doi: 10.1109/ACCESS.2019.2919956.

[13] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," Applied Soft Computing, vol. 49, pp. 248–258, Dec. 2016, doi: 10.1016/j.asoc.2016.08.011.

[14] M. Alzaqebah et al., "Hybrid Feature Selection Method based on Particle Swarm Optimization and Adaptive local Search Method," International Journal of Electrical and Computer Engineering, vol. 11, Dec. 2020, doi: 10.11591/ijece.v11i3.pp2414-2422.

[15] J. Feng and Z. Gong, "A Novel Feature Selection Method With Neighborhood Rough Set and Improved Particle Swarm Optimization," IEEE Access, vol. 10, pp. 33301–33312, 2022, doi: 10.1109/ACCESS.2022.3162074.

[16] K. Bouzoubaa, Y. Taher, and B. Nsiri, "Predicting DOS-DDOS Attacks: Review and Evaluation Study of Feature Selection Methods based on Wrapper Process," IJACSA, vol. 12, no. 5, 2021, doi: 10.14569/IJACSA.2021.0120517.

[17] R. P. Lippmann et al., "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, Hilton Head, SC, USA: IEEE Comput. Soc, 1999, pp. 12–26. doi: 10.1109/DISCEX.2000.821506.

[18] S. Revathi and D. A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," International Journal of Engineering Research, vol. 2, no. 12, 2013.

[19] A. Husain, A. Salem, C. Jim, and G. Dimitoglou, "Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset," in 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Dec. 2019, pp. 1–7.

[20] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A Survey on Ensemble Learning for Data Stream Classification," ACM Comput. Surv., vol. 50, no. 2, p. 23:1-23:36, Mar. 2017, doi: 10.1145/3054925.

[21] P. Singh and V. Ranga, "Attack and intrusion detection in cloud computing using an ensemble learning approach," Int. j. inf. tecnol., vol. 13, no. 2, pp. 565–571, Apr. 2021, doi: 10.1007/s41870-020-00583-w.

[22] Md. K. Islam, P. Hridi, Md. S. Hossain, and H. S. Narman, "Network Anomaly Detection Using LightGBM: A Gradient Boosting Classifier," in 2020 30th International Telecommunication Networks and Applications Conference (ITNAC), Nov. 2020, pp. 1–7. doi: 10.1109/ITNAC50341.2020.9315049.

[23] A. Singh, K. Chatterjee, and S. C. Satapathy, "An edge based hybrid intrusion detection framework for mobile edge computing," Complex Intell. Syst., vol. 8, no. 5, pp. 3719–3746, Oct. 2022, doi: 10.1007/s40747-021-00498-4.

[24] N. G. Bhuvaneswari Amma and P. Valarmathi, "ORaBaN: an optimized radial basis neuro framework for anomaly detection in large networks," Int. j. inf. tecnol., vol. 14, no. 5, pp. 2497–2503, Aug. 2022, doi: 10.1007/s41870-022-00991-0.

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 2 | Pages 367-374 | e-ISSN: 2347-470X

[25] M. A. Ferrag, L. Shu, H. Djallel, and K.-K. R. Choo, "Deep Learning-Based Intrusion Detection for Distributed Denial of Service Attack in Agriculture 4.0," Electronics, vol. 10, no. 11, Art. no. 11, Jan. 2021, doi: 10.3390/electronics10111257.

[26] S. Mahadik, P. M. Pawar, and R. Muthalagu, "Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT)," J Netw Syst Manage, vol. 31, no. 1, p. 2, Oct. 2022, doi: 10.1007/s10922-022-09697-x.