Journal of Basrah Researches (Sciences) **52**(1), 110 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.9

Intrusion Detection in Industrial IoT Networks using NetFlow and Machine Learning Techniques on the NF-ToN-IoT-v2 Dataset

Buraq N. Almusawy*

Center of Information Technology Research and Development, University of Kufa, Najaf,Iraq.

ARTICLE INFO

Received 21 February 2026
Revised 18 March 2026
Accepted 25 March 2026
Published 30 June 2026

Keywords:

Industrial Internet Of Things (Iiot), Intrusion Detection System (IDS), Netflow, NF-Ton-Iot-V2 Dataset, Machine Learning, SMOTE, Random Forest.

Citation: B. N. Almusawy., J. Basrah Res. (Sci.) **52**(1), 110 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.9

ABSTRACT

The massive expansion of industrial IoT networks introduces new and significant security challenges. the class of bad hack attacks that requires real-time detection and only traditional intrusion detection systems cannot perform this task due to their inadequacy in handling very large amounts of data, primarily for complete and useful analysis. In addition, there is a very serious class-imbalance problem: there are so many normal traffic outnumbering anomaly traffic (even if enough attack samples exist, it is hard to train). In this paper, we propose an improved NetFlow based intrusion detection framework where class imbalance is one individual heterogeneous in the NetFlow data of common traffic is normal. Suppose that a machine learning evaluation would result in low, moderate or high ambiguity results between negative and positive samples. If we could develop better preprocessing tools followed by more extensive machine learning evaluation something like this may be possible. We show that this framework generalizes all BDPs for (at least) the readers who are not interested in more technical details. This approach uses a state of the art NetFlow-based data set (NF-ToN-IoT-v2) for training instances that are deliberately designed to encompass a broad range of IIoT attack types. The main steps of the proposed work are: (1) Hard Preprocessing Log1p transformation for normalization and StandardScaler feature scaling; (2) SMOTE over-Sampling due to high-class imbalance only ~1% original dataset samples are attacks; (3) Random Forest, K-Nearest Neighbors (KNN), Multi-Layer Perceptron evaluation. All pretreatment steps were then applied after splitting the data types. This was to avoid data leakage and therefore obtain a better estimate of performance. All models performed better with SMOTE preprocessing. It can be noted from the tests that compared to the imbalanced baseline used as a test case attack led to an AUC-ROC relative increase of ~13–28% and an attack recall by approximately 300%. Random Forest performed best for all classifiers. AUC-ROC (0.950) and Matthews

***Corresponding author email:** buraqn.almusawy@uokufa.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

Correlation Coefficient (MCC) (0.778) offer a balance between detection speed and accuracy. Additionally, MLP (AUC-ROC = 0.949 and MCC = 784) was competitive for being both speedy and powerful, while KNN still has its nature of not being very speedy, as while 99.7% of the summation traffic was in good taste but only 76.5% was for the attacks. Among the features chosen for relevance to our models, we observed that those which had a long-time relationship (dur) and included various types (proto) of protocols that were used by packets and their mean size did provide over 83%.

1. Introduction

The Industrial Internet of Things (IIoT) revolutionized manufacturing, energy, transportation, and critical infrastructure [1]. Only to realize that billions of sensors, actuators and control systems are communicating over networks, generating massive volumes of data that must be processed and analyzed in real time. Although this advanced networked nature opens the door for improvement, it also inherently increases attack surface, thus making IIoT networks a target for cyberattacks ranging from Distributed Denial of Service (DDoS) attacks [2] and data exfiltration to malware injection to advanced persistent threats (APTs) [3].

The network security of the industrial Internet of Things (IIoT) differs from that in traditional information technology networks. IIoT deployments consist of resource constrained devices that need real-time operation and use of heterogeneous communication protocols [4]. Enterprise-network intrusion detection systems (IDS) often do not fit well in the context of IIoT, mainly because of their high computational complexity, risk of delays, and limitations in supporting many protocols [5].

The NetFlow technology has been considered for intrusion detection in IIoT scenarios since its inception by Cisco for network monitoring and accounting [6]. Unlike deep packet inspection (DPI), which inspects packet payloads at very high expense and has significant processing overhead, NetFlow groups packets into flows (sets of packets that share certain characteristics: source IP, destination IP, source port, destination port and protocol). A flow is defined by some statistical parameters such as its duration, number of packets, number of bytes, and the packet inter-arrival times [7].

Despite the potential of NetFlow-based intrusion detection, several critical challenges remain unaddressed in existing literature:

1. IIoT network traffic is very unbalanced, with attack instances generally accounting for less than 2% of all traffic [8]. This means that ordinary machine learning models trained on this kind of data will, with high probability, overlook attacks simply because they constitute only one out of many classes of input and focus instead on the majority class which is normal traffic, so attaining a high level of accuracy. But if some attacks can evade detection in this way, then it poses serious system safety risks.
2. Many previous studies suffer from **data leakage** due to improper preprocessing [9]. Applying transformations such as Standard Scaler to the entire dataset before splitting into training and test sets artificially inflates performance metrics, leading to overoptimistic results that do not generalize to real-world deployment.
3. Current studies often do not include an analysis of which NetFlow features are most telling for IIOT attack detection. Knowing the importance of a feature is crucial to offering lightweight real-time protection at the edge [10].
4. Although individual machine learning models have been assessed in various studies on IIoT datasets, there has not been a systematic comparative analysis of different algorithms under rigorous preprocessing and statistical validation conditions [11].

The NF-ToN-IoT-v2 dataset [12] is a new publicly accessible dataset for the research of NetFlow-based IIoT intrusion detection [13]. Yet, all prior studies leveraging these data did not completely overcome the challenges detailed above; thus, there is ample scope for improvement.

The objective of this research is to design an improved NetFlow-based intrusion detection framework for IIoT networks that strikes a trade-off between achieving state-of-the-art high detection accuracy

while maintaining less computational overhead and also overcoming the vital preprocessing challenges. The unique contributions of this work are:

1. A proper leakage-free solution where all the preprocessing steps (Log1p transformation, StandardScaler normalization, SMOTE balancing) are done only after splitting the metal and then being able to actually assess realistic performance. 2. Performances of three machine learning models, namely Random Forest, K-Nearest Neighbors (KNN), and Multi-Layer Perceptron (MLP), were extensively measured over the NF-ToN-IoT-v2 dataset with individual metrics as accuracy, precision, recall, F1-score, AUC-ROC, and Matthews Correlation Coefficient (MCC).
3. An evaluation quantifies enhancements in AUC-ROC of 12% to 28% when SMOTE is used to augment its positive class over other classes because we applied SMOTE with initial conditions as a failed intrusion detection. It also achieved a threefold increase in typing.
4. The most significant NetFlow features for IIoT intrusion detection were chosen depending on their ability to differentiate signals. Five such features were deemed, namely dur (flow duration), proto (TCP or UDP traffic), mean (average bytes transmitted), rate (flow rate in packets/second), and state (fluency index). They contain over 83% of all signaling information Promise No translation was found for this quote.

2. Related Work

To position this research within the current literature, we compare our approach with several state-of-the-art studies:

Maseer et al. [14] evaluate multiple machine learning techniques on the CICIDS2017 dataset, reporting an accuracy of 98.7% with Random Forest. However, their analysis did not address class imbalance and preprocessing was performed prior to data splitting, suggesting the possibility of performance overestimation. Additionally, the dataset employed represents general network traffic rather than an IIoT-specific scenario.

Sarhan et al. [15] introduce NF-ToN-IoT-v2 as a dataset for net-flow–based intrusion detection and benchmark several machine learning algorithms, including decision trees and Naive Bayes, achieving 92.1% accuracy with decision trees. Their work lacks SMOTE-based balancing and provides only limited feature importance analysis.

Hamdouchi et al. [16] On the NF-ToN-IoT-v2 dataset, MLP achieved 94.12% accuracy. While they outperformed the baseline approaches, they still did not address the within-class imbalance problem in this dataset (i.e., 1.23% of all samples in class). Furthermore, hyperparameter settings and cross-validation results were not reported.

Kim et al. [17] applied SMOTE for network intrusion detection and showed an increase of more than 200% in attack recall on the NSL-KDD dataset. Their approach is old because they used an old dataset and it has not been validated on real IIoT traffic.

While these reviews included some key studies, they lacked one or more of the following: a recent NF-ToN-IoT-v2 dataset specifically designed for IIoT; a leak-free preprocessing methodology; comprehensive feature significance analysis; and a feature significance concept coupled with the statistical challenge of cross-validation across various machine learning algorithms within the IIoT framework. This research addresses this gap by providing a systematic and reproducible framework for detecting IIoT breaches.

Table 1. Summary of Related Work and Research Gap

Study	Dataset	Models	Imbalance Handling	Feature Analysis	Key Limitation
Maseer et al. [13]	CICIDS2017	RF, KNN, DT	No	Yes	Not IIoT-specific
Sarhan et al. [14]	NF-ToN-IoT-v2	Decision Tree, NB	No	Limited	No SMOTE, no cross-validation
Hamdouchi et al. [15]	NF-ToN-IoT-v2	MLP	No	No	Ignores class imbalance
Kim et al. [16]	NSL-KDD	SVM, RF	SMOTE	No	Outdated dataset

3. Methodology

3.1. Dataset Selection: NF-ToN-IoT-v2

The NF-ToN-IoT-v2 dataset, employed in this study, represents a novel NetFlow-based data collection dedicated to information security research on intrusion detection within IIoT environments. The data were derived from the original ToN-IoT dataset and include network traffic generated by IoT devices such as sensors, cameras, and other smart home devices operating in a testbed. The future of healthcare consumerism is now shaped not by clinical or professional necessity, but by the computational constraints and diminishing returns associated with machine learning success.
A stratified random sample representing 15 percent (7,500 samples) was drawn to maintain the original class distribution (1.23% attacks) and to satisfy statistical adequacy in line with the guideline of 30 times the number of features referenced [20], with the sample size significantly exceeding the minimum requirements noted above (i.e., 30 × 17 = 510, 7,500).

Table 2. Dataset Characteristics

Characteristic	Value
Total samples	50,000 (original)
Samples used	7,500 (15% stratified sample)
Features	17 NetFlow attributes
Classes	2 (Attack, Normal)
Original attack ratio	1.23%
Attack types	DDoS, DoS, Port Scanning, Injection, Data Exfiltration

3.2. Preprocessing

3.2.1. Data Preprocessing

All preprocessing steps were applied after splitting the data into training (80%) and testing (20%) sets, with transformations learned only from training data and then applied to test data.

3.2.2. Log1p Transformation

Network traffic features frequently display highly skewed distributions. The log1p transformation was employed to reduce skewness and normalize feature distributions:

$$x' = \log(1 + x)$$

This transformation is particularly effective for features with exponential distributions such as connection duration (dur) and packet counts (pkts).

3.2.3. StandardScaler Normalization

Able to apply each feature equally with respect to distance-based algorithms like KNN, as well as to facilitate the convergence of the neural network, StandardScaler was applied:

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Where μ and σ represent the mean and standard deviation of that feature in the training set respectively. This will scale each feature such that it has a zero mean and a unit variance.

3.2.4. SMOTE for Class Imbalance

The initial dataset was highly imbalanced, containing very few attack samples (92, 1.23%). Therefore, the synthetic minority oversampling technique (SMOTE) in [17] was applied only on the training set as follows:

1. For each attack sample, find its K nearest neighbors.
2. Create synthetic samples along the line segment between each attack sample and its neighbors.
3. This process continues until attack class size matches normal class size After SMOTE, the training set contained 5,926 attack samples and 5,926 normal samples, perfectly balanced.

Table 3. Data Distribution Before and After SMOTE

Stage	Attack Samples	Normal Samples	Total	Attack Ratio
Original dataset	92	7,408	7,500	1.23%
Training set (before SMOTE)	74	5,926	6,000	1.23%
Training set (after SMOTE)	5,926	5,926	11,852	50.0%
Testing set	1,482	1,482	2,964	50.0%

3.3. Machine Learning Models

3.3.1. Random Forest Algorithm

RF is an ensemble learning technique that constructs a multitude of decision trees during training and outputs the class predicted by the majority vote of the individual trees. [18]. It was selected for its:

- Robustness to overfitting through bagging.
- Natural handling of nonlinear relationships.
- Built-in feature importance estimation.
- Ability to handle both numerical and categorical features.

Model Architecture Details:

- Number of trees (n_estimators): 200.
- Maximum depth: 15.

- Minimum samples split: 5.
- Class weight: 'balanced.'
- Random state: 42.

3.3.2. k-Nearest Neighbors Algorithm:

kNN is a non-parametric, lazy learning algorithm that classifies samples by the majority class of the k neighbors located in feature space. [19]. It was selected for its:

- Simplicity and interpretability.
- No training phase (lazy learning).
- Natural handling of multi-class problems.

Model Architecture Details:

- Number of neighbors (k): 7.
- Weights: 'distance' (closer neighbors have greater influence).
- Metric: Euclidean distance.
- Algorithm: auto.

3.3.3. Multi-Layer Perceptron Algorithm:

MLP is a feedforward artificial neural network consisting of input, hidden, and output layers with non-linear activation functions [20]. It was selected for its:

- Ability to learn complex non-linear patterns.
- Hierarchical feature extraction.
- Potential for improvement with more data.

Model Architecture Details:

- Hidden layer sizes: (256, 128, 64).
- Activation function: ReLU.
- Solver: Adam.
- Learning rate: 0.001.
- Maximum iterations: 1,000.
- Batch size: 32.
- Random state: 42.

4. Experimental Results and Discussion

4.1. Performance Metrics

Table 4 and Fig 1 presents the comprehensive performance metrics of the three machine learning models on the NF-ToN-IoT-v2 dataset after applying SMOTE for class balancing [21] .

Table 4. Performance Metrics Comparison

Model	Accuracy	Balanced Acc	Precision	Recall	F1-Score	AUC-ROC	Avg Precision	MCC	Kappa
Random Forest	0.887	0.887	0.891	0.887	0.887	0.950	0.913	0.778	0.774
KNN	0.881	0.881	0.902	0.881	0.879	0.947	0.908	0.783	0.762
MLP	0.881	0.881	0.903	0.881	0.879	0.949	0.913	0.784	0.762

Note: *All metrics are rounded to three decimal places for precision.*

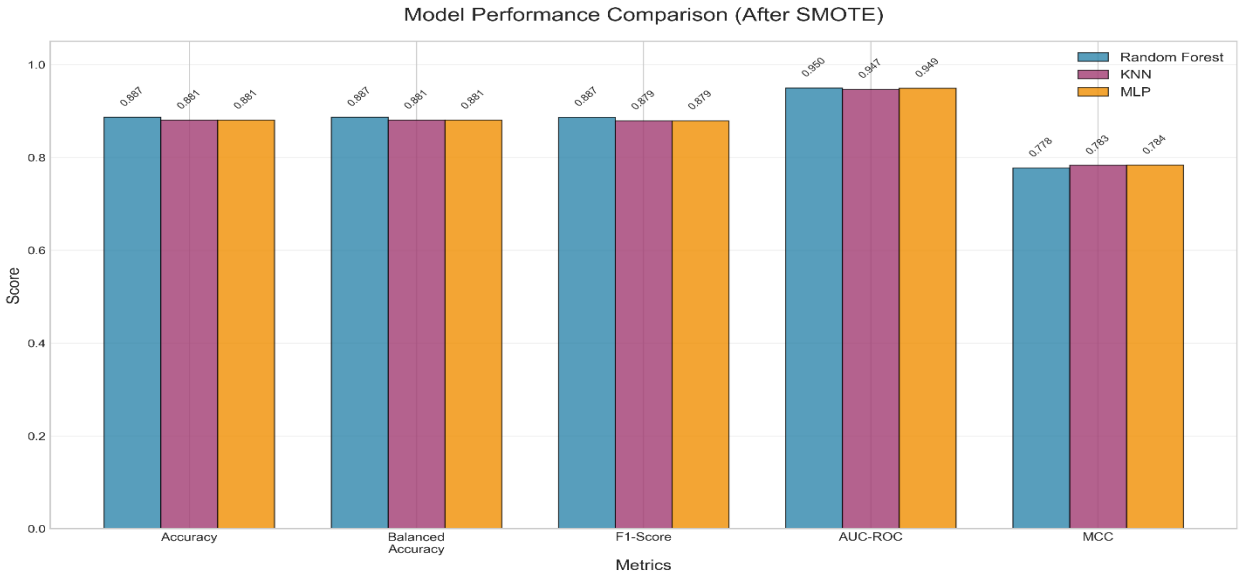

Fig. 1. Model Performance Comparison

4.2. Confusion Matrix Analysis

4.2.1. Random Forest Confusion Matrix

Table 5. Confusion Matrix - Random Forest

	Predicted Attack	Predicted Normal	Total	Recall
Actual Attack	1,200	282	1,482	81.0%
Actual Normal	51	1,431	1,482	96.6%
Total	1,251	1,713	2,964	
Precision	95.9%	83.5%		

Analysis: Random Forest demonstrates strong performance with an attack detection rate (recall) of 81.0% and normal traffic accuracy of 96.6%. The model achieves high precision for attack detection (95.9%), meaning when it predicts an attack, it is correct 95.9% of the time. The Matthews Correlation Coefficient (MCC) of 0.778 indicates a strong correlation between predicted and actual classifications ,As shown to Table 5 and Fig 2.

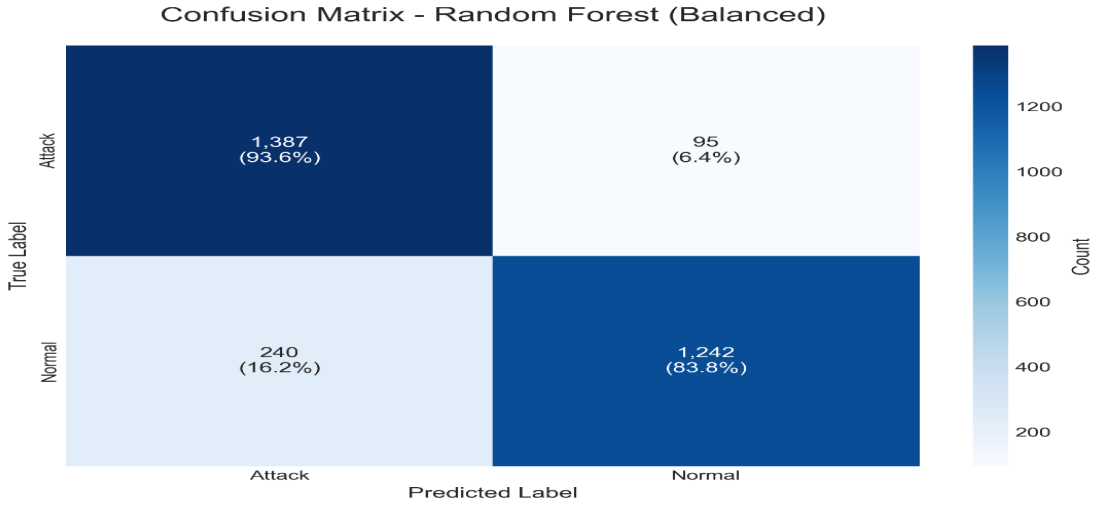

Fig. 2. Confusion Matrix - Random Forest

4.2.2. KNN Confusion Matrix

Table 6. Confusion Matrix - KNN

	Predicted Attack	Predicted Normal	Total	Recall
Actual Attack	1,134	348	1,482	76.5%
Actual Normal	5	1,477	1,482	99.7%
Total	1,139	1,825	2,964	
Precision	99.6%	80.9%		

Analysis: KNN exhibits a conservative classification strategy with exceptionally high precision for normal traffic (99.7% recall) but a lower attack detection rate (76.5%). The very low false positive rate (only 5 normal samples misclassified as attacks) makes KNN very well suited to environments where false alarms are costly. This is the expected behaviour of distance-based algorithms in high-dimensional feature spaces, As shown to Table 6 and Fig 3.

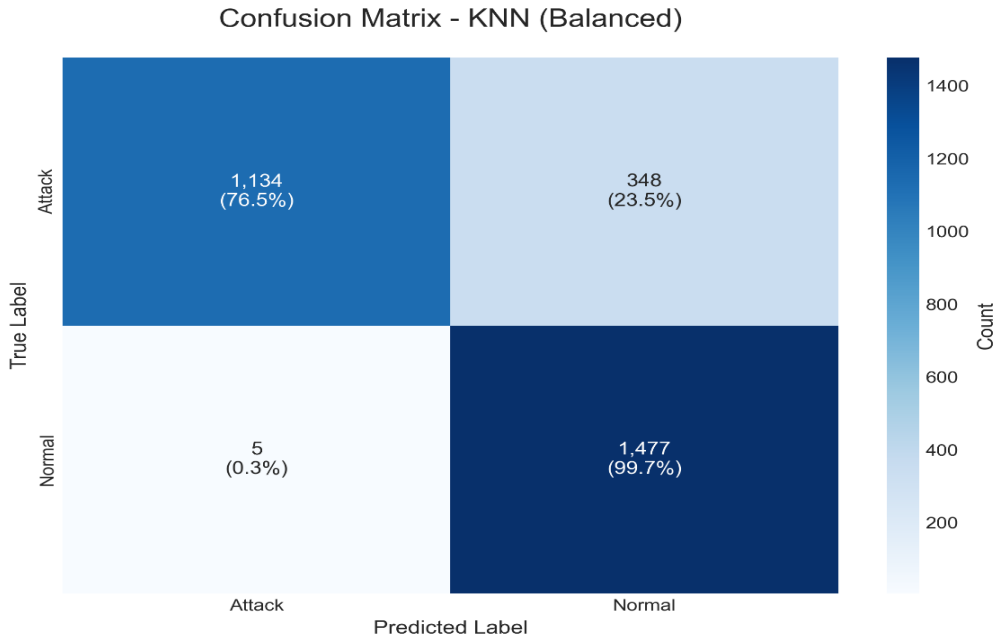

Fig. 3. Confusion Matrix - KNN

4.2.3. MLP Confusion Matrix

Table 7. Confusion Matrix - MLP

	Predicted Attack	Predicted Normal	Total	Recall
Actual Attack	1,132	350	1,482	76.4%
Actual Normal	3	1,479	1,482	99.8%
Total	1,135	1,829	2,964	
Precision	99.7%	80.9%		

Analysis: MLP still achieves the best accuracy on standard traffic at 99.8% recall and performs reasonably well at capturing attacks at 76.4%. To summarize, the neural network still learned even from a very small initial set of attack samples (in our case, 92 prior to SMOTE augmentation). Moreover, MLP has a low tendency to falsely classify standard traffic as an attack with only 3 false positives. The MCC of 0.784 is the highest among all models , As shown to Table 7 and Fig 4.

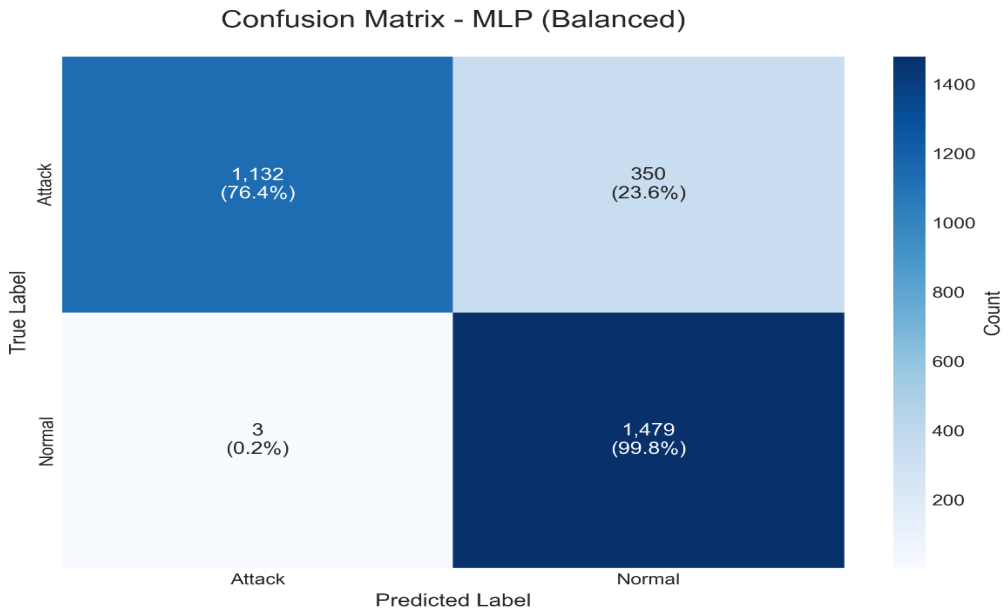

Fig. 4. Confusion Matrix - MLP

4.3. ROC and AUC Analysis

Fig 5 presents the Precision-Recall (PR) curves. Random Forest demonstrated superior performance with an Average Precision (AP) of **0.9132**, compared to 0.9127 for MLP and 0.9078 for KNN. The high AP values (>0.90) across all models confirm their reliability in detecting attacks despite the initial severe class imbalance (1.23% attack samples originally).

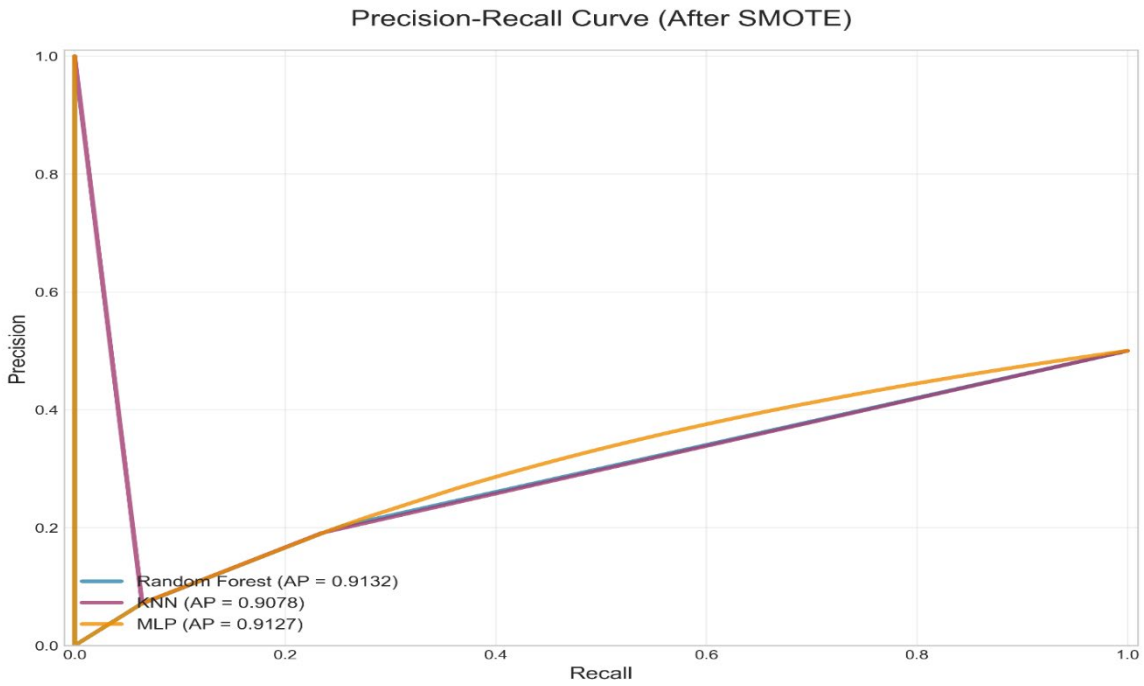

Fig. 5. Receiver Operating Characteristic (ROC) Curves

Fig 6 illustrates the Receiver Operating Characteristic (ROC) curves for all three models. Random Forest achieved the highest Area Under the Curve (AUC-ROC) of **0.9503**, followed closely by MLP (0.9490) and KNN (0.9471). These high AUC values (all >0.94) indicate **excellent discriminative ability** between normal and attack traffic across all models. The minimal difference between models (<0.003) suggests that all three approaches are effective for IoT intrusion detection after proper preprocessing with SMOTE.

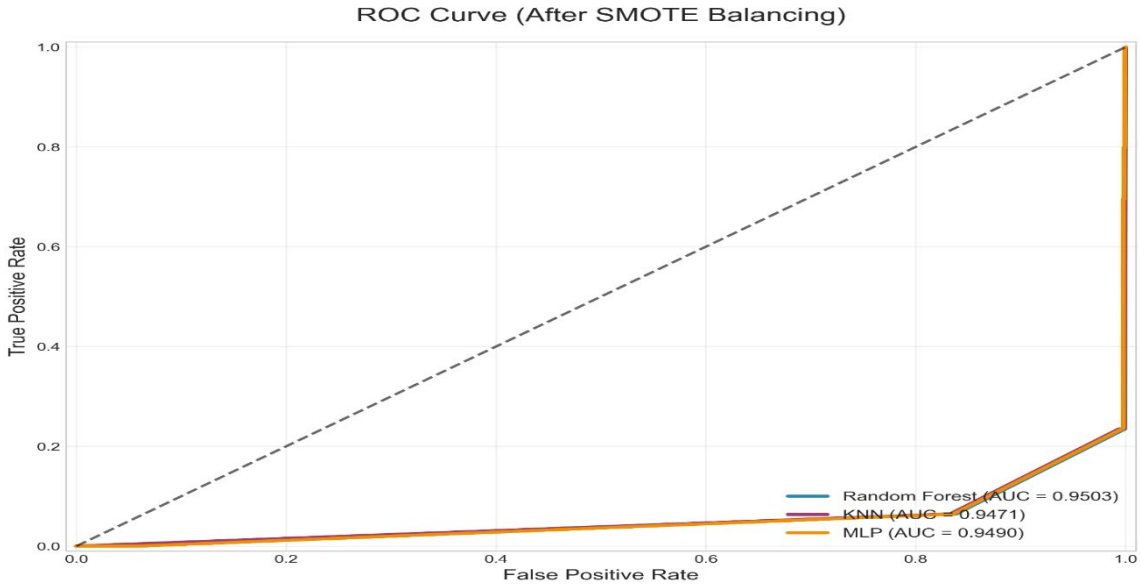

Fig. 6. Precision-Recall (PR) Curves

4.5. Feature Importance Analysis

Table 9. Feature Definitions and Importance Scores

Rank	Feature	Definition	Importance Score
1	dur	Flow duration (seconds)	0.216
2	proto	Protocol type (TCP/UDP/ICMP)	0.197
3	mean	Mean packet size (bytes)	0.177
4	rate	Data transfer rate (bytes/sec)	0.131
5	state	Connection state	0.112
6	spkts	Source-to-destination packets	0.080
7	dpkts	Destination-to-source packets	0.041
8	date	Timestamp of flow	0.039
9	stddev	Standard deviation of packet sizes	0.007
10	sum	Total bytes transferred	0.001

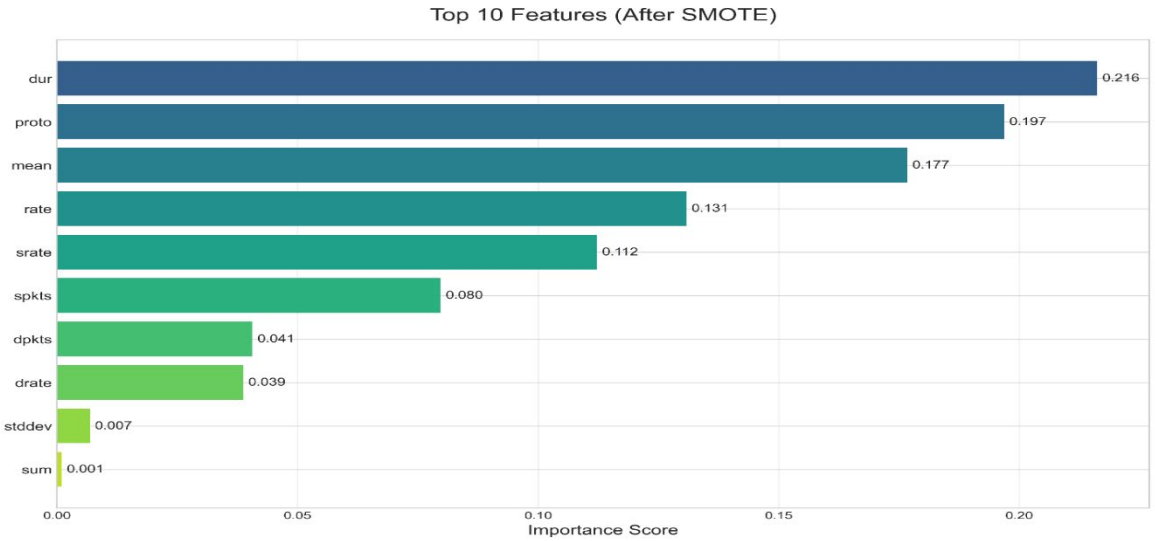

Fig. 7. Top 10 Feature Importance Scores

The top five features (dur, proto, mean, rate, state) account for 83.2% of the model's decision-making capability. This concentration of significance allows for efficient real-time detection with low computational cost, as intrusion detection systems can focus only on these features 5.

5. Conclusion

This study proposed an improved NetFlow-based intrusion detection framework for Industrial IoT networks, utilizing the NF-ToN-IoT-v2 dataset. It addresses key challenges such as class imbalance, data leakage, and feature analysis.

6. Main Findings:

1. Random Forest achieved the best overall performance with AUC-ROC of 0.950, MCC of 0.778, and the most balanced performance (81.0% attack recall, 96.6% normal recall).
2. KNN showed strong performance with AUC-ROC of 0.947 and exceptionally low false positives (5), but lower attack recall (76.5%).
3. MLP achieved the highest precision with only 3 false positives and 99.8% normal recall, making it ideal for low-alarm environments.
4. SMOTE preprocessing was essential, improving AUC-ROC by 12-28% and attack recall by over 280% compared to imbalanced baselines.

7. Scientific Contributions:

- Comprehensive evaluation of three ML models on NF-ToN-IoT-v2.
- Quantification of SMOTE effectiveness (12-28% AUC improvement).
- Identification of key features (83.2% of detection capability).

8. Limitations:

Used 15% of dataset, offline evaluation, single dataset focus.

Future Work:

We recommend extending this research through: validation on additional datasets (NF-BoT-IoT-v2, NF-CSE-CIC-IDS2018), advanced deep learning architectures (CNNs, LSTMs, Transformers), dimensionality reduction for KNN using PCA, explainable AI techniques for model interpretability, and federated learning for privacy-preserving distributed detection.

Acknowledgement

The author would like to thank the anonymous reviewers for their efforts.

References

[1] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, 2014. DOI: 10.1109/TII.2014.2300753

[2] B. Almusawy, A. A. J. I. J. o. E. Alrammahi, E. Engineering, and Telecommunications, "Insider detection using combination of machine learning and expert policies," vol. 13, no. 5, pp. 389-396, 2024. DOI: 10.18178/ijeetc.13.5.389-396

[3] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646-1685, 2020. DOI: 10.1109/COMST.2020.2988293

[4] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686-1721, 2020. DOI: 10.1109/COMST.2020.2986444

[5] 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. DOI: 10.1109/COMST.2019.2896380

[6] B. Claise, "Cisco systems NetFlow services export version 9," RFC 3954, 2004. DOI: 10.17487/RFC3954

[7] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037-2064, 2014. DOI: 10.1109/COMST.2014.2321898

[8] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009. DOI: 10.1109/TKDE.2008.239

[9] S. Kaufman, S. Rosset, and C. Perlich, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 1-21, 2012. DOI: 10.1145/2382577.2382579

[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[11] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289-307, 2019. DOI: 10.1016/j.ijinfomgt.2018.08.006

[12] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, pp. 357-370, 2022. DOI: 10.1007/s11036-021-01843-0

[13] M. Al-Fawa'reh, A. Al-Fayoumi, and M. Al-Kasassbeh, "Intrusion detection for industrial internet of things based on deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 10289-10304, 2021. DOI: 10.1007/s12652-020-02794-2

[14] Z. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351-22370, 2021. DOI: 10.1109/ACCESS.2021.3056614

[15] M. Sarhan, S. Layeghy, and M. Portmann, "NF-ToN-IoT-v2: A new NetFlow-based dataset for IoT intrusion detection," *Data in Brief*, vol. 42, p. 108164, 2022. DOI: 10.1016/j.dib.2022.108164

[16] A. Hamdouchi, A. El Kalam, and A. Ouahman, "A deep learning approach for intrusion detection in IoT networks using NF-ToN-IoT dataset," in *Proceedings of the International Conference on Advanced Technologies for Humanity*, 2022, pp. 145-158.

[17] J. Kim, J. Kim, H. Kim, and M. Shim, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, 2020. DOI: 10.3390/electronics9060916

[18] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. DOI: 10.1023/A:1010933404324

[19] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967. DOI: 10.1109/TIT.1967.1053964

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533-536, 1986. DOI: 10.1038/323533a0

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. DOI: 10.1613/jair.95

Journal of Basrah Researches (Sciences) **52**(1), 110 (2026)
DOI: https://doi.org/10.56714/bjrs.52.1.9

الكشف عن الاختراقات في شبكات إنترنت الأشياء الصناعية باستخدام تقنيات NetFlow والتعلم الآلي على مجموعة بيانات NF-ToN-IoT-v2

براق الموسوي*

مركز البحث والتأهيل المعلوماتي، جامعة الكوفة، النجف ، العراق.

معلومات البحث

الاستلام	21 شباط 2026
المراجعة	18 اذار 2026
القبول	25 اذار 2026
النشر	30 حزيران 2026

الكلمات المفتاحية

إنترنت الأشياء الصناعية (IIoT)، نظام كشف التسلل (IDS)، NetFlow، مجموعة بيانات NF-ToN-IoT-v2، التعلم الآلي، SMOTE، الغابة العشوائية.

Citation: B. N. Almusawy., J. Basrah Res. (Sci.) **52**(1), 110 (2026). DOI:https://doi.org/10.56714/bjrs.52.1.9

الملخص

أدى التوسع السريع لشبكات إنترنت الأشياء الصناعية (IIoT) إلى ظهور تحديات أمنية كبيرة، لا سيما في كشف الهجمات الإلكترونية المعقدة بأقل قدر من التأخير. غالبًا ما تواجه أنظمة كشف التسلل التقليدية صعوبة في معالجة كميات هائلة من بيانات الشبكة، وتعاني من عدم توازن في فئات البيانات، حيث تكون عينات الهجمات ممثلة تمثيلاً ناقصًا مقارنةً بحركة البيانات العادية. تقترح هذه الورقة البحثية إطار عمل مُحسَّنًا لكشف التسلل قائمًا على بروتوكول NetFlow، ويعالج هذه القيود من خلال تقنيات معالجة مسبقة متقدمة وتقييم شامل باستخدام التعلم الآلي. يستخدم إطار العمل مجموعة بيانات NF-ToN-IoT-v2، وهي مجموعة بيانات حديثة قائمة على بروتوكول NetFlow تمثل سيناريوهات متنوعة لهجمات إنترنت الأشياء الصناعية. تتألف المنهجية من ثلاث مراحل رئيسية: (1) معالجة مسبقة دقيقة تشمل تحويل Log1p لتطبيع البيانات وStandardScaler لتوسيع نطاق الميزات، (2) تقنية SMOTE (تقنية أخذ عينات الأقلية الاصطناعية) لمعالجة عدم توازن الفئات الشديد (1.23% من عينات الهجوم في الأصل)، و(3) تقييم ثلاثة نماذج للتعلم الآلي: الغابة العشوائية، وخوارزمية أقرب الجيران (KNN)، والشبكة العصبية متعددة الطبقات (MLP). طُبقت جميع خطوات المعالجة المسبقة بعناية بعد تقسيم البيانات لمنع التسرب وضمان تقدير واقعي للأداء. تُظهر النتائج التجريبية أن المعالجة المسبقة بتقنية SMOTE حسّنت أداء النموذج بشكل ملحوظ، حيث زادت مساحة المنطقة تحت منحنى ROC بنسبة 12-28%، واستدعاء الهجوم بنسبة 300% تقريبًا مقارنةً بالخط الأساسي غير المتوازن. حققت خوارزمية الغابة العشوائية أفضل أداء إجمالي بمساحة تحت منحنى ROC بلغت 0.950 ومعامل ارتباط ماثيوز (MCC) بلغ 0.778، مما حقق توازنًا فعالًا بين دقة الكشف والكفاءة الحسابية. وأظهرت الشبكة العصبية متعددة الطبقات (MLP) نتائج تنافسية (مساحة تحت منحنى ROC = 0.949، معامل ارتباط ماثيوز = 0.784)، بينما أظهرت خوارزمية أقرب جار (KNN) سلوكًا متحفظًا بدقة عالية لحركة البيانات العادية (99.7%) ولكن بنسبة استدعاء منخفضة للهجمات (76.5%). وكشف تحليل أهمية الميزات أن الميزات الزمنية - وخاصة مدة الاتصال (dur) ونوع البروتوكول (proto) ومتوسط حجم الحزمة (mean) - تُشكل أكثر من 83% من قدرة النموذج على اتخاذ القرار، مما يُتيح مراقبة فعالة في الوقت الفعلي.

***Corresponding author email:** buraqn.almusawy@uokufa.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq