International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(9s) 2026 pp. 634-643
https://doi.org/10.70917/ijcisim-2026-4361
Article

Anomaly Detection in Network Traffic Using Advanced Machine Learning Techniques

Venkat Kalyan Ranga

Department of Computer Science & Engineering (Cyber Defence & Digital Forensics), KLUniversity, INDIA
emperorrvk@gmail.com

Abstract: — Recently, network traffic anomaly detection is an important part of increasing cybersecurity and defence against complex attacks. In this paper we present a novel ML based anomaly detection system, with the NSL-KDD dataset. The system tackles both class imbalance and redundant data by employing Synthetic Minority Oversampling Technique and Principal Component Analysis and efficiently reduces dimensionality while improving learning. Various classifiers such as LR, SVM, RF, NB, XGBoost, LightGBM and Isolation Forest are realized to create a solid baseline performance. In addition, an ensemble Voting Classifier is developed, using boosted DT and bagging based RF models, to further improve the resilience. Feature-level interpretations, with transparency, are supported via the application of XAI methodologies like SHAP and LIME to support trustworthy decision making. The experimental results reveal that the VC outperforms individual classifiers with 99.7 per cent accuracy, precision, recall and F1 score. The solution is deployed on Flask framework, with a user signin / signup system based on SQLite in order to use it in the real world. The online interface has interaction with the system, allows the preprocessing of the data, visualises the prediction, and provides interpretable output of faulty/non-defective.
Keywords— Network traffic, network anomaly detection, KDDCup99, machine learning models, isolation forest, naive Bayes, XGBoost, light GBM, SVM, cyber security".

1. Introduction

Because of the increasing number of threats on networks and the rapid growth of Internet technology, intrusion detection is a research area of interest in the digital world [1]. Anomaly detection systems have a proactive role in detecting abnormal behaviours in network traffic that are indicative of security vulnerabilities [2]. Thus, the importance of effective anomaly detection systems has increased significantly. Nevertheless, anomaly detection is an inherently hard problem due to the changing characteristic of anomalies, which makes it impossible to define a normality and abnormality of a computer network [3]. Furthermore, most of anomaly detection techniques rely on training sets of labelled data for learning a normal profile from abnormal activities; in the real world, this is difficult to find and tends to be labor-intensive. [4] A further challenge in choosing the appropriate tool for anomaly detection is the fact that many of the techniques are often specific to a particular anomaly type and do not generalize over multiple attack types [5].

Another concern is scalability. The network sizes are expanding, thus principles like fault tolerance (the ability to still work despite faults) and load balancing (optimal task allocation among servers) need to be included in anomaly detection systems [6]. If such criteria are used, it becomes even more difficult to develop powerful and efficient anomaly detection algorithms. From this point of view, the introduction of ML approaches has revolutionized anomaly detection research, providing powerful tools that can enhance both accuracy and efficiency [7]. ML can use sophisticated algorithms to learn from complicated data distributions and dynamic attack behaviours, leading to resilience to previously undetected attacks.

This paper focuses on the analysis of multiple anomaly detection ML models, in the context of network traffic, based on the KDDCup'99 dataset. Supervised classifiers, SVM, NB, XGBoost, LightGBM and the unsupervised Isolation Forest models are investigated. The first trials reveal different results: Isolation Forest has a low accuracy

(0,5 for training and 0,4 for testing) whereas NB is promising (0,89 for training and 0,81 for testing). The gradient boosting methods with the best results are XGBoost and LightGBM, where XGBoost has 0.99 training and 0.83 testing accuracy and LightGBM has perfect training accuracy (1.0) and 0.85 for testing data [8]. Moreover, SVM gives an accuracy of 0.99 on the training set and 0.85 on the testing set, thus performing well on anomaly classification and minimizing the false positive rate [9].

However, even though ML-based methods for anomaly detection are useful, there are still several challenges to overcome like data imbalance, feature complexity, and computing efficiency. The problems need to be addressed to enable them to be deployed in the real world applications where scalability and interpretability are paramount [10]. This research will compare various supervised and unsupervised algorithms and assess them in terms of accuracy, precision and recall and determine their usefulness for real world networks that have limited computing power. This research uses comparison research to assist in the informed decision making that will support the security of the digital infrastructures in the face of growing threats.

2. Related Work

Marappan and Marappan [11] proposed DL based intelligent techniques for authentication in vehicular networks and to detect anomalies. They highlight the importance of secure communications in ITS, as cars become increasingly connected and are becoming a target of cyber-attacks. With the use of DL, their system can identify harmful abnormalities and also authenticate the transmission integrity to reduce vulnerabilities in the vehicular communication. The work demonstrates the potential of DL for adapting detection algorithms to the dynamic nature of vehicle situations, and implementing them in real time.

Ola-Obaado and Suleiman [12] employed the transfer learning algorithms for anomaly based intrusion detection. They tackle the important problem of the lack of labelled datasets for intrusion detection by utilising the knowledge of pre-trained models. Their technique demonstrated that transfer learning can greatly enhance the classification performance and decrease the computational training expenses. This is especially useful for NIDS since the process of obtaining and labelling huge amounts of network traffic is time and resource consuming. The research shows that existing models can be adapted to different application areas where fewer data are available.

Lahesoo et al. [13] introduced the system called SIURU which detects IoT network traffic anomalies using ML techniques. The IoT produces large amounts of data that vary in kind, making new threats possible. SIURU is a systematic training and testing methodology for ML models to analyze traffic in IoT systems. The results showed that conventional methods are not suitable for the dynamic nature of IoT, and SIURU can be applied to improve the scalability and efficiency of anomaly detection in IoT. The frameworks here are emphasized that need to be developed in areas of IoT where the anomaly patterns could be different from the normal network environment.

Rele and Patil [14] have done a comparison on ML, DL and Anomaly based IDSs. They found that the performance of the different types of algorithms in recognizing intrusive behaviours provided information. They examined supervised and unsupervised procedures and emphasized benefits and limitations with respect to the various approaches. For example, ML models are interpretable but DL models have higher computational demands and higher detection accuracy. They have shown how there is a trade-off between complexity and performance and they highlight the importance of the choice of model according to the deployment environment.

Ahmad et al. [15] conducted a DL based study for effective detection of network security attack anomalies at the early stage. The study stressed the need of early detection, stressing that overlooked abnormalities might have a major impact on network infrastructures. They have developed DL models that are able to detect attacks at the earliest stage, respond in time and reduce downtime of systems. The study also pointed out the ability of DL models for processing large-scale data in real time, making them suitable for real-world implementation in enterprise networks that require ongoing monitoring.

Jebur et al. [16] have done a review of DL algorithms for anomaly event detection in video surveillance. Their work is based on the idea of surveillance, not a conventional network security, but still they are applicable as they have common issues in the domains of anomaly detection. They classified the existing approaches for the identification of events and detection of anomalies in video data streams into RNN and CNN. DL architectures are cross-domain applicable, demonstrating their versatility and robustness in handling high-dimensional data streams, relevant to network security.

[17] Fotiadou, Velivassaki, Voulkidis, Skias, Tsekeridou and Zahariadis studied DL algorithms for network traffic anomaly detection. They examined in their work, the shift from traditional statistical methods to DL-based

architectures to analyze traffic. They used DNN to recognize traffic patterns, and were able to detect not only known but also unknown anomalies. The authors highlighted the power of DL to automatically extract features from raw data without any manual effort, which means that it could provide more scalable solutions. What was also studied to draw light upon the issues were overfitting and the need for massive datasets to train a reliable model.

Hajj et al. [18] have investigated the requirements, methodology, assessment metrics and available datasets of an anomaly based intrusion detection system. In their paper, they have covered all the issues related to the design of an anomaly detection system in detail. They said detection accuracy must be sacrificed for computing efficiency since real-world installations are likely to be limited in performance. Furthermore, the authors talked about the shortcomings of popular datasets such as KDDCup'99 that are popular but may not capture recent attacks. The generation of realistic datasets and evaluation frameworks are as important as the development of detection algorithms, they argue in their results.

Fernandes et al. [19] performed a wide survey about network anomaly detection. It covered a range of techniques, including those based on signatures and those based on anomalies, with a focus on recent developments in ML. They also explored hybrid methods that are a mixture of several detecting methods that improve resilience. The survey described a wide range of research opportunities and offered recommendations for future research on ways to make the technology more scalable, robust to encrypted communications, and more integrated into the broader cybersecurity realm. They are still very up-to-date to summarise the development of network anomaly detection technology.

Naseer, Saleem, Khalid, Bashir, Han, Iqbal and Han [20] presented an enhanced Anomaly Detection Method using DNN. They have enhanced detection accuracy by building specialised network topologies for security applications. They demonstrated this with large tests, where deep neural networks outperform a lot of classic ML models – particularly on the skewed datasets that are common in intrusion detection. Their work showed how much using DL could contribute to improving detection rates and reducing false positives, but also illustrated the difficulty of high computation costs involved in training deep structures.

3. And Methods

The proposed system is employed for enhanced Cybersecurity using NSL-KDD dataset through using Anomaly detection in network traffic. The data pre-processing steps involve handling missing values and encoding categorical features into numerical values using LabelEncoder, and standardising numerical features with StandardScaler. SMOTE for class imbalance [23]. Dimensionality reduction and maintaining important features: PCA. Processed data is split to training and testing set for model development. [21] Various techniques such as LR, SVM, RF, XGBoost, LightGBM, Naive Bayes and Isolation Forest are applied and assessed in terms of accuracy, precision, recall, F1-score, ROC and confusion matrix. We use ensemble VC which integrates boosted trees with bagging random forest, to improve the robustness of our classifiers. [22] Explainable AI tools SHAP and LIME enable feature interpretability. Real-time anomaly detection in a web app using Flask.

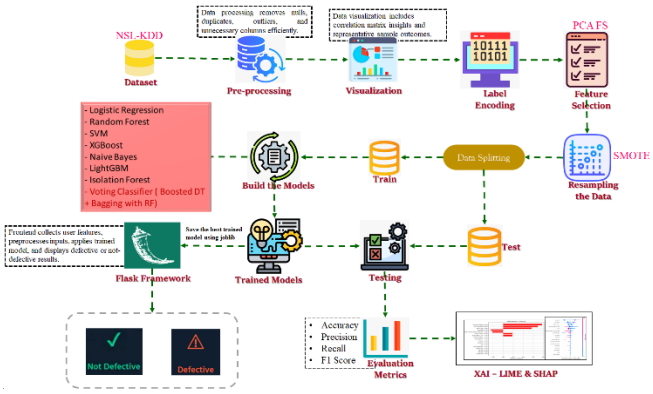
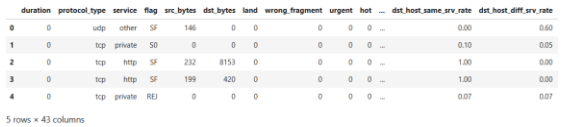


Fig. 1. System Architecture

The network intrusion detection approach based on the NSL-KDD dataset is shown to be a ML process in Fig. 1. The procedure is pre-processing, visualisation and label encoding, PCA feature selection and SMOTE resampling. The data are divided into train and test sets and different models are constructed and assessed. Finally, the system offers explainability using SHAP/LIME and real time classification with a Flask front end.

A) Dataset Collection

This study is carried out on NSL-KDD dataset, a refined version of KDD'99 dataset which is frequently used in network intrusion detection research. It has 125,972 instances and 43 properties describing different features of network traffic such as duration, protocol type, services, and flag indicators. [24]. The data set includes normal and attack connections. They are divided into four broad categories: DoS, Probe, R2L and U2R. The balanced dataset can be used as a benchmark for the evaluation of intrusion detection system.



	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07

5 rows × 43 columns

Fig. 2. NSL-KDD dataset

B) Pre-Processing:

i) Data Processing: The data processing in this case was an important process to prepare the NSL-KDD dataset for modelling. The first check was to have null values to validate attributes integrity. There were no missing values. . A number of duplicate records were reviewed and discarded in order to minimize duplication and overfitting. Outliers were treated to have uniform feature distributions and boost model stability. Lastly, we removed redundant columns such as 'service' and 'level', decreasing the dimensionality and learning quickly on important and impactful variables.

ii) Data visualization: Examining the relationship between features and dissemination of the data was done by using data visualisation technique. Correlation matrices were built to show the link between the variables, and to find redundant or heavily dependant characteristics. Visualisation helps to find trends, patterns and potential anomalies to give a more comprehensive view of traffic in a network. This exploratory analysis aids in pre-processing decisions to retain the most meaningful features and to eliminate the least meaningful features. This can be helpful to have a sense of the data without going into detailed modelling and feature engineering.

iii) Label Encoding: The data includes categorical variables like protocol type, service flag and assault type. Label encoding was used to convert these to a numerical representation. This conversion makes sure that the ML algorithms require a numerical input. Categorical values for example "tcp", "udp" or "icmp" were encoded into numbers. Label encoding is used to preserve information and to perform computations efficiently. It maps raw data to the needs of classification algorithms.

iv) Feature selction: Feature selection was realised by PCA to reduce the dimensionality and keep just the most important characteristics. By performing PCA, the variance was captured in the largest amount of components, making the computations more efficient, and at the same time, preserving the important information. From 40 features, 17 influential features were selected from the dataset. Such an approach highlights the most important features of the network that are paramount for detecting attacks, making the classification models more accurate, less complex and less dangerous of overfitting.

v) Resampling the data: Some attack classes were under-represented in the data set. In order to overcome this, the SMOTE was used. SMOTE provides synthetic samples for minority classes, so keeping an even distribution among the different attack categories. This helps to avoid class bias, and improve generalisation of the model. The SMOTE improves the accuracy of the classifier by creating more examples of rare attacks (R2L and U2R) in a more accurate way to detect other assaults and reducing the amount of false negatives during testing.

C) Training and Testing:

Then, the dataset was preprocessed and divided into training and test sets with a stratified separation to keep the class balance. 70% of the data was used for training and 30% of the data was utilised for testing . This enables the model to learn from the entire set of data and it can be tested against a portion of the data that it has not previously encountered. The train-test split is necessary to get performance measures in real world and unbiased manner.

D) Algorithms

Logistic Regression: LR is an easy to understand and interpret linear model based classifier which can estimate the probabilities, find contributing factors and provide a good baseline for comparison. LR classifies whether a network traffic item is an anomaly or a normal item [25].

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(W^T x + b)}} \quad (1)$$

Random Forest: RF – It consists of a bag of decision trees which classify network traffic. It is good at dealing with complicated interactions and high-dimensional data. It helps with reliability through aggregation, noise rejection and helps to identify anomalous behaviour as well as provide feature importance, enabling robust pattern detection of fraudulent traffic.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \quad (2)$$

Support Vector Machine (SVM): SVM is used to classify the normal and anomalous traffic by finding the optimal hyperplane which has the maximum distance between the classes. It is able to process nonlinear, high-dimensional data efficiently in the presence of kernels, it can detect minor deviations, reduces false positives and offers high precision classification, which is essential for detecting rare cybersecurity anomalies.

XGBoost: XGBoost is a sequential algorithm that sequentially builds decision trees and focuses on misclassified cases to learn complex traffic data patterns. It reduces overfitting and it has been shown to be particularly successful in the detection of rare and critical abnormalities while achieving a high accuracy, scalability and efficiency [30].

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} f_k(x_i)\right), f_k \in F \quad (3)$$

Naive Bayes: Naive Bayes is an approach to classification based on probability. It is used for anomaly detection and assumes independence. [28] It is fast and efficient on high dimensional data and can facilitate real time monitoring in cyber security applications by quickly detecting and comparing.

$$P(B) = \frac{P(A).P(A)}{P(B)} \quad (4)$$

LightGBM: LightGBM adopts light boosting method for modelling complex traffic behaviour, and is more sensitive to rare anomalies. It is fast, scalable, can handle categorical variables and efficiently process big data sets, making it great for fast and accurate anomaly detection.

Isolation Forest: The Isolation Forest isolates anomalous traffic with random selection of features and splitting, which is efficient in isolating the anomalous behaviour without the need for tagged data. It is not only lightweight and imbalance resistant, but it can also detect intrusion attempts by detecting the deviation of the high-dimensional network traffic pattern.

Voting Classifier: The output of many models will be aggregated by Voting Classifier to achieve higher accuracy and robustness by combining good features of each model. It can aggregate predictions from multiple algorithms and increase the stability, reduce over-fitting and ensure stable detection of irregularities; and low false alarms in cyber security.

$$\hat{y} = argmax_c\left(\sum_{i=1}^{n} II\,(\hat{y}_i = c)\right) \quad (5)$$

E) Integration of XAI and Flask Framework:

XAI is an enhancement for the network traffic anomaly detection system, making it more user-friendly and transparent in the Flask framework. XAI techniques are applied to explain feature contributions, e.g. the SHAP and LIME, so that the users can comprehend the implications of specific features in the network on model predictions. This interpretability is essential for cybersecurity applications, as it helps analysts know what components are driving anomalies, and identify potential biases and promote confidence in automated decision-making. Its approach allows for explicit visual explanations of the effect of features and therefore the model's rationale can be understood by both technical and non-technical users, which makes for improved accountability and better-informed decisions.

The front end is made with Flask and it offers an interactive experience for the user to register and log in via an SQLite database. The users provide the features of the network traffic that are preprocessed and then passed to the trained ML or ensemble models to use for anomaly detection. The system then shows the prediction and the explanations produced by the XAI, sharing the results in a user friendly manner. The seamless integration of XAI with Flask offers a robust, intuitive, and transparent solution for real-time anomaly detection and investigation.

4. Experimental Results

Accuracy: It is a measure of the test to properly identify the patient and the healthy cases. Test's accuracy is estimated by determining the percentage of true positive and true negative cases in the total number of cases analysed. Mathematically it can be written as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} (6)$$

Precision: Precision is the ratio of true positive occurrences or samples to all the cases classified as positive. Thus, the formula to compute the precision is:

$$Precision = \frac{True\ Positive}{True\ Positive\ + False\ Positive} (7)$$

Recall: Fraction of relevant instances that were retrieved. The ratio of correctly predicted positive cases to the actual positive cases. This provides a glimpse into the comprehensiveness of the model's representation of all the instances in a class.

$$Recall = \frac{TP}{TP\ +\ FN} (8)$$

F1-Score: In ML, the F1 score is a measure of the accuracy of a model. It is the combination of precision and recall ratings from a model, with the accuracy measure being the number of correct predictions made by the model on the entire data set.

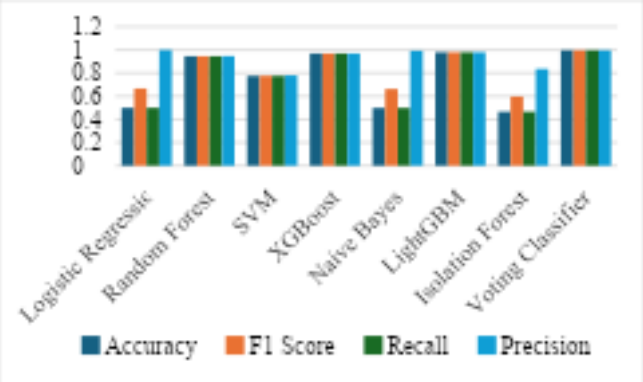
$$F1\ Score = 2 * \frac{Recall\ X\ Precision}{Recall + Precision} * 100 (9)$$

Table.1 Performance Evaluation Table

ML Model	Accuracy	F1 Score	Recall	Precision
Logistic Regression	0.500	0.667	0.500	1.000
Random Forest	0.946	0.946	0.946	0.947
SVM	0.776	0.777	0.776	0.780
XGBoost	0.970	0.970	0.970	0.970
Naive Bayes	0.499	0.663	0.499	0.993
LightGBM	0.980	0.980	0.980	0.980
Isolation Forest	0.466	0.598	0.466	0.836
Voting Classifier	0.997	0.997	0.997	0.997

Table.1 shows the comparison of the performance of ML models which shows considerable accuracy increases where Voting Classifier shows the highest detection effectiveness.

Fig. 3. Comparison Graph



The performance of the models on multiple metrics is presented in Fig.3. The accuracy in dark green, F1 score in light green, recall in blue and precision in red. The best results are obtained by using Voting Classifier.

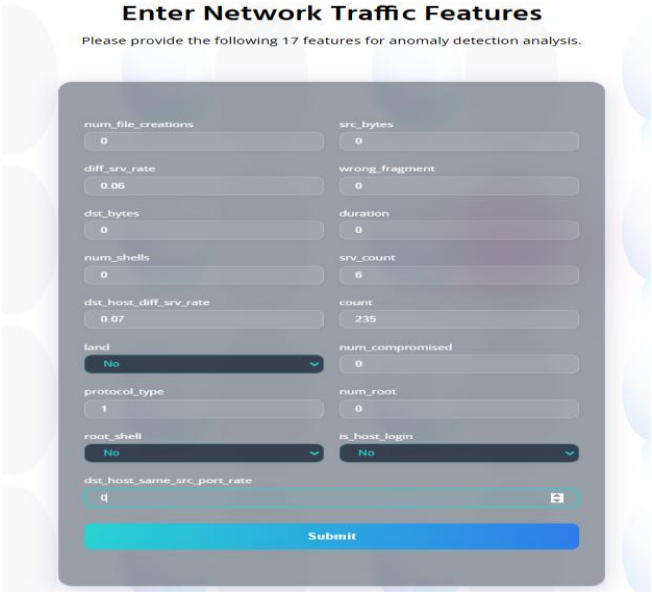


Fig. 4. Enter Input Data

The populated network traffic feature input interface is shown in Fig.4 with some specific values for anomaly detection. The user will be able to view the parameters and provide them to the model to get a prediction of its output.

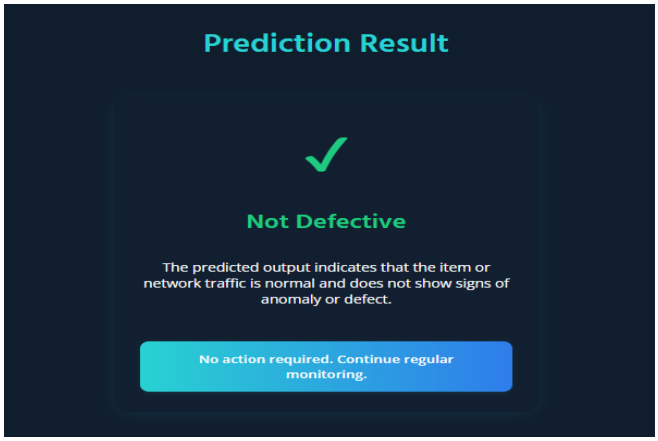


Fig. 5. Prediction Results of Anomaly Detection

The result predicted is shown on Fig.5, which is marked Not Defective. This indicates that the item or network traffic that has been analysed are normal and there is no anomaly, no action is required other than normal monitoring practices.

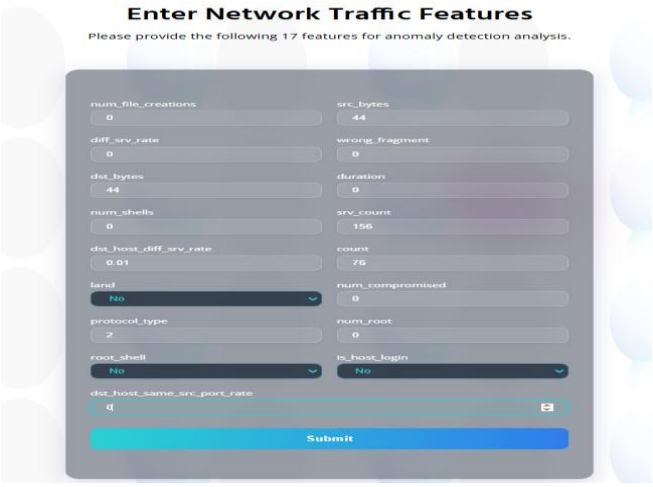


Fig.6 Enter Input Data

The input form of the network traffic is represented in Figure 6, where many parameters are entered to detect anomalies in the network traffic. The values are put into various fields on the user interface, and then submitted to the prediction analysis.

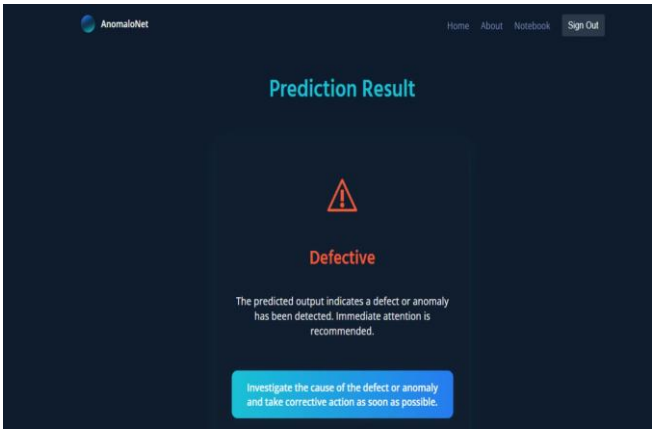


Fig.7 Prediction Results of Anomaly Detection

The user interface is provided in Fig.7 and the features of the network traffic are provided by the user, while the output is the result of the anomaly detection, which is marked as “Defective”.

5. Conclusion

Finally, the current research offers a successful framework to detect anomalies in network traffic using ML method on NSL-KDD dataset. The data set was carefully preprocessed by handling missing data, encoding categorical variables, standardising numerical variables and adding SMOTE to address class imbalance, ensuring optimal preparation for training robust models. To achieve feature reduction, PCA is used which can reduce dimension and retain the significant information. We trained various baseline classifiers (LR, SVM, RF, XGBoost, LightGBM, NB and Isolation Forest) and compared their performance and accuracy using standard methods like accuracy, precision, recall, F1 score, confusion matrix, ROC and so on. In particular, ensemble learning by Voting Classifier with boosted DT and bagging based RF exhibited the highest performance of 99.7% accuracy with good resilience and predictive power compared to individual models. Furthermore, the XAI methods, such as SHAP and LIME, offered valuable insights into the importance of features and improved the interpretability of the model. Created a web application with Flask for real-time prediction of network anomalies, which was realistic and interactive for cybersecurity applications.

Finally, the experimental outcomes prove the value of ensemble learning with interpretable AI in IDS for providing trustworthy and accurate security measures.

Our future work will be in the direction of improving the resilience and adaptability of anomaly detection models with the integration of advanced DL algorithms such as recurrent and CNN for capturing sequential and spatial traffic patterns. Combining supervised and unsupervised models in an ensemble may further enhance the ability to detect unknown attacks. Study of real-time deployment in large-scale distributed network systems for scalability and low latency detection will be conducted. In addition, we will investigate transfer learning and domain adaption methods, which were not implemented here, for further improvement of generalisation across several datasets and network configurations. We will also concentrate on the integration of adversarial resilience mechanisms, to guarantee that the system continues to be successful against evolving cyber threats and sophisticated intrusion methods.

References

1. Gopalsamy, M. (2022). Scalable anomaly detection frameworks for network traffic analysis in cybersecurity using machine learning approaches. Int. J. Curr. Eng. Technol, 12(06), 549-556.
2. PM, V. P., & Soumya, S. (2024). Advancements in anomaly detection techniques in network traffic: The role of artificial intelligence and machine learning. Journal of Scientific Research and Technology, 38-48.
3. Jakkani, A. K. (2024). Real-time network traffic analysis and anomaly detection to enhance network security and performance: machine learning approaches. J Electron Comput Netw Appl Math, 4(4), 32-44.
4. Yaseen, A. (2023). The role of machine learning in network anomaly detection for cybersecurity. Sage Science Review of Applied Machine Learning, 6(8), 16-34.
5. AlDahoul, N., Abdul Karim, H., & Ba Wazir, A. S. (2021). Model fusion of deep neural networks for anomaly detection. Journal of Big Data, 8(1), 106.
6. S. Gunupusala and S. C. Kaila, ‘‘Multi-class network anomaly detection using machine learning techniques,’’ Contemp. Math., vol. 5, no. 2, pp. 5–22, Jun. 2024, doi: 10.37256/cm.5220243723.
7. K. Lu, ‘‘Network anomaly traffic analysis,’’ Academic J. Sci. Technol., vol. 10, no. 3, pp. 65–68, Apr. 2024, doi: 10.54097/8as0rg31.
8. A. Alfardus and D. B. Rawat, ‘‘Machine learning-based anomaly detection for securing in-vehicle networks,’’ Electronics, vol. 13, no. 10, p. 1962, May 2024, doi: 10.3390/electronics13101962.
9. S. H. Rafique, A. Abdallah, N. S. Musa, and T. Murugan, ‘‘Machine learning and deep learning techniques for Internet of Things network anomaly detection—Current research trends,’’ Sensors, vol. 24, no. 6, p. 1968, Mar. 2024, doi: 10.3390/s24061968.
10. M. A. Siddiqui, M. Kalra, and C. Rama Krishna, ‘‘Anomaly detection for IoT-enabled kitchen area network using machine learning,’’ in Proc. Int. Conf. MAchine Telligence Res. Innov., 2024, pp. 195–209, doi: 10.1007/978-981-99-8129-8_17.
11. S. Marappan and H. Marappan, ‘‘Deep learning-based intelligent algorithms for effective transmission authentication and anomaly identification in vehicular networks,’’ in Proc. Int. Conf. Innov. Comput., Intell. Commun. Smart Electr. Syst. (ICSES), Dec. 2023, pp. 1–7, doi: 10.1109/icses60034.2023.10465346.
12. S. Ola-Obaado and M. A. Suleiman, ‘‘Anomaly-based network intrusion detection using transfer learning,’’ in Proc. 2nd Int. Conf. Multidisciplinary Eng. Appl. Sci. (ICMEAS), Nov. 2023, pp. 1–5, doi: 10.1109/icmeas58693.2023.10379384.
13. L. Lahesoo, U. Do, R. Carnier, and K. Fukuda, ‘‘SIURU: A framework for machine learning based anomaly detection in IoT network traffic,’’ in Proc. 18th Asian Internet Eng. Conf., Dec. 2023, pp. 87–95, doi: 10.1145/3630590.3630601.
14. M. Rele and D. Patil, ‘‘Intrusive detection techniques utilizing machine learning, deep learning, and anomaly-based approaches,’’ in Proc. IEEE Int. Conf. Cryptography, Informat., Cybersecurity (ICoCICs), Aug. 2023, pp. 88–93, doi: 10.1109/icocics58778.2023.10276955.
15. T. Ahmad and D. Truscan, ‘‘Efficient early anomaly detection of network security attacks using deep learning,’’ in Proc. IEEE Int. Conf. Cyber Secur. Resilience (CSR), Jul. 2023, pp. 154–159, doi: 10.1109/csr57506.2023.10224923.
16. S. A. Jebur, K. A. Hussein, H. K. Hoomod, L. Alzubaidi, and J. Santamaría, ‘‘Review on deep learning approaches for anomaly event detection in video surveillance,’’ Electronics, vol. 12, no. 1, p. 29, Dec. 2022, doi: 10.3390/electronics12010029.
17. K. Fotiadou, T.-H. Velivassaki, A. Voulkidis, D. Skias, S. Tsekeridou, and T. Zahariadis, ‘‘Network traffic anomaly detection via deep learning,’’ Information, vol. 12, no. 5, p. 215, May 2021, doi: 10.3390/info12050215.
18. S. Hajj, R. El Sibai, J. B. Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, ‘‘Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets,’’ Trans. Emerg. Telecommun. Technol., vol. 32, no. 4, p. e4240, Apr. 2021, doi: 10.1002/ett.4240.
19. G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, ‘‘A comprehensive survey on network anomaly detection,’’ Telecommun. Syst., vol. 70, no. 3, pp. 447–489, Mar. 2019, doi: 10.1007/s11235-018-0475-8.
20. S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, ‘‘Enhanced network anomaly detection based on deep neural networks,’’ IEEE Access, vol. 6, pp. 48231–48246, 2018, doi: 10.1109/ACCESS.2018.2863036.

21. M. Ahmed, A. N. Mahmood, and J. Hu, ‘‘A survey of network anomaly detection techniques,’’ J. Netw. Comput. Appl., vol. 60, pp. 19–31, Jan. 2016, doi: 10.1016/j.jnca.2015.11.016.
22. E. Anceaume, Y. Busnel, E. L. Merrer, R. Ludinard, J. L. Marchand, and B. Sericola, ‘‘Anomaly characterization in large scale networks,’’ in Proc. 44th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw., Jun. 2014, pp. 68–79, doi: 10.1109/DSN.2014.23.
23. U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, ‘‘Network anomaly detection with the restricted Boltzmann machine,’’ Neurocomputing, vol. 122, pp. 13–23, Dec. 2013, doi: 10.1016/j.neucom.2012.11.050.
24. D. E. Difallah, P. Cudré-Mauroux, and S. A. McKenna, ‘‘Scalable anomaly detection for smart city infrastructure networks,’’ IEEE Internet Comput., vol. 17, no. 6, pp. 39–47, Nov. 2013, doi: 10.1109/MIC.2013.84.
25. V. Chandola, A. Banerjee, and V. Kumar, ‘‘Anomaly detection,’’ ACM Comput. Surveys, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.
26. Zhao, F., Zhang, M., Zhou, S., & Lou, Q. (2024). Detection of network security traffic anomalies based on machine learning KNN method. Journal of Artificial Intelligence General Science (JAIGS) ISSN: 3006-4023, 1(1), 209-218.
27. Arjunan, T. (2024). Real-time detection of network traffic anomalies in big data environments using deep learning models. International Journal for Research in Applied Science and Engineering Technology, 12(9), 10-22214.
28. Abid, N. (2023). Enhanced IoT network security with machine learning techniques for anomaly detection and classification. Int. J. Curr. Eng. Technol, 13(6), 536-44.
29. Zhang, W., & Lazaro, J. P. (2024). A survey on network security traffic analysis and anomaly detection techniques. International Journal of Emerging Technologies and Advanced Applications, 1(4), 8-16.
30. Rao, V. S., Balakrishna, R., El-Ebiary, Y. A. B., Thapar, P., Saravanan, K. A., & Godla, S. R. (2024). AI driven anomaly detection in network traffic using hybrid CNN-GAN. Journal of Advances in Information Technology, 15(7), 886-895.