INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Semi-Supervised Intrusion Detection Based on Stacking and Feature-Engineering to Handle Data Imbalance

A Sagaya Priya[1]*, **S Britto Ramesh Kumar**[2]

1 Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University, Trichy, Tamil Nadu, India
2 Department of Computer Science, St. Joseph's College (Autonomous), Affiliated to Bharathidasan University, Trichy, Tamil Nadu, India

OPEN ACCESS

Received: 17-09-2022
Accepted: 04-11-2022
Published: 14-12-2022

Citation: Priya AS, Kumar SBR (2022) Semi-Supervised Intrusion Detection Based on Stacking and Feature-Engineering to Handle Data Imbalance. Indian Journal of Science and Technology 15(46): 2548-2554. https://doi.org/10.17485/IJST/v15i46.1885

* **Corresponding author**.

rschlrsagayapriya@outlook.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To design an architecture that can effectively handle the imbalance levels and complexities in the network data to provide qualitative predictions. **Methods:** Experiments were performed with KDD CUP 99 dataset, NSL- KDD dataset and UNSW- NB15 dataset. Comparisons were performed with SAVAER-DNN model. Oversampling technique is used for data balancing, and the stacking architecture handles the issue of overtraining introduced due to oversampling. **Findings:** The proposed Stacking and Feature engineering-based Semi-supervised (SFS) model presents a combined architecture that integrates data balancing, feature engineering and a stacking-based prediction model that balances data to reduce imbalance, reduces the data size, and also provides highly effective predictions. **Results:** indicate 2% increase in accuracy levels on the UNSW-NB15 dataset and 10% increase in accuracy levels in the NSL-KDD dataset. **Novelty:** The architecture has been designed in a domain-specific manner. Multiple intrusion detection datasets, each with different levels of imbalance, have been used to depict the generic nature of the SFS model.

Keywords: Intrusion Detection; Data Imbalance; Stacking; Feature Engineering; Oversampling; Semi Supervised Learning

1 Introduction

Real time network data is generally laden with imbalance. Hence, an effective model that can handle varied levels of imbalance is required to operate on real time data [(1)]. An integrated approach that uses genetic based grey wolf optimization algorithms and feature selection algorithm has been proposed by [(2)]. This work enhances the existing grey wolf optimization algorithm and integrates it with genetic algorithm to provide an enhanced prediction model for network intrusion detection. The model is highly complex in nature and is not capable of handling imbalance. An intrusion reduction system specifically designed for handling Distributed Denial Of Service (DDoS) attacks has been proposed by Baklinil [(3)]. A deep learning based model for intrusion detection

has been proposed by Fu [4]. A similar model using deep learning based technique for intrusion detection has been proposed by Rani [5]. An attack-based intrusion detection model for enhanced detection over imbalanced data has been proposed by Pimsarn [6]. This work uses a sliding window technique and combines it with a logical fractal dimension to provide effective detection of DDoS attacks. The window size is obtained automatically and the model also evaluates predictions using different hyper parameters to show the efficiency of the detection technique. The window based models have been designed with static window sizes, which are obtained initially. They tend to fail when the data distribution varies due to changes in data in due course of time.

A collaborative model to detect network intrusions has been proposed by Guarascio et al [7]. This work is mainly based on creating collaboration among the detection models to create better and more improved overall intrusion detection system. It defines an ecosystem that performs knowledge sharing to improve the prediction accuracy. Other similar words dealing with providing specifications for collaborative intrusion detection includes Structured Threat Information CybereXpression (STIX) by Jordan [8], XGBOOST based model by L [9], and Trusted Automated exchange of Indicator Information (TAXII) by [10]. A secure intrusion detection system for MANET has been proposed by Prasad [11]. A Random Forest model incorporating SMOTE for data balancing has been proposed by Wu [12]. This technique provides an integrated architecture that includes imbalance handling and intrusion detection modules. Random forest has the issue of over training, which has to be handled independently over large data. This work however does not consider this aspect during the model creation process. Using grouping techniques for intrusion detection using clustering based models has also gained prominence. Some grouping based techniques to perform intrusion detection include works by Siddiqui [13] and Mehmood [14]. Although collaborative models are highly effective, they are not generic in nature and has to be fine-tuned according to the data to ensure high predictions.

Deep learning model that uses regularisation the best auto encoder to perform network intrusion detection has been proposed by Yang [15]. Other similar techniques for intrusion detection include a MapReduce based model by Asif [16] and feature engineering based model by Yao [17]. Deep learning models are computationally complex in nature, hence prediction times are usually high.

Overall analysis indicates that most models do not handle the issue of data imbalance, which introduces large amount of bias in the prediction process. Further, the voluminous nature of the data results in huge time requirements when operated upon highly complex models like deep learning techniques. Most of the models were observed to be data driven in nature, as the model is fine-tuned for the specific data in hand. Even slight variations in the data can result in model failure. This work presents the Stacking and Feature engineering based Semi supervised (SFS) machine learning model for intrusion detection in a networked scenario. The technique also uses a balancing module to handle the imbalance. Feature engineering reduces the size of data to reduce the computational complexity of the model. The stacking approach combined with semi supervised learning provides enhanced predictions. Further, the model is generic in nature and has been designed in a domain specific manner, rather than being data specific. Performance over the varied datasets validates the domain centric nature of the model. A combination of these techniques ensures that the issues discussed in the literature works are handled effectively.

2 Methodology

The intrusion detection process is highly complicated by the multi-class nature of data and the data imbalance contained in network transactions [18,19].This work presents a highly suitable architecture for classifying data containing varied imbalance levels. The model has been designed to incorporate components that can handle data imbalance and reduce the data's complexity by introducing feature engineering-based techniques. The model has been designed to be generic, and the domain-centric nature of operations enables enhanced predictions.

2.1 Stacking and Feature Engineering Based Semi Supervised (SFS Intrusion Detection

The proposed architecture combines stacking and feature engineering approaches and uses semi-supervised prediction processes for training. The proposed model, Stacking and Feature Engineering based Semi-supervised (SFS) intrusion detection system, comprises four modules. The initial module performs data pre-processing and data balancing, the next module performs feature selection, the next module performs the Semi supervised first level predictions, and the final module performs the final prediction. The algorithm for the SFS model is provided below.

Algorithm-1 Stacking and Feature Engineering based Semi Supervised IDS

Input: Imbalanced data (KDD CUP 99, NSL-KDD, UNSW-NB15)

Output: Predictions on imbalanced data

1. Input network transmission data

2. Data preprocessing to perform encoding and remove inconsistencies
3. Identify the imbalance level
4. For each additional majority record contained in the data
    (a) Random select two minority instances
    (b) Generate new instance based on the mean value of the selected instances
5. Use decision tree to identify the entropy values of the features
6. Based on the entropy perform feature selection
7. Create multiple data subsets by sampling with replacement
8. Create multiple instances of Gaussian Mixture and Decision Tree models
9. For each created model
    (a) Pass a distinct data subset for training
10. Pass the entire training data to the trained models for prediction
11. Integrate predictions with the class label to create level 2 training data
12. Pass the level 2 training data to Logistic Regression model for training
13. For each instance i in test data
    (a) Pass i to all the base learners
    (b) Integrate the predictions
    (c) Pass the integrated predictions to trained Logistic Regression model
    (d) Obtain final predictions

2.2 Data Pre-processing and Balancing

Intrusion detection data from multiple datasets are used as the training data to analyze the efficiency and the generic operability of the proposed SFS model. Intrusion detection data is composed of features obtained from network transmissions. These features are required to be analyzed in order to improve the qualitative nature of the training data. Data analysis shows that the data comprises categorical, string, and numerical attributes. Machine learning models can directly use numerical attributes. However, categorical and string attributes should be analyzed prior to usage. Categorical attributes are generally converted to numerical attributes using encoding techniques. This work uses one hot encoding as the preferred technique. The string attributes are eliminated. Some datasets contain class attributes represented in categorical formats. This attribute describes the type of transmission as normal or anomalous. Some data sets represent this attribute in a multi-class format representing the type of anomalous traffic. This work considers the classification process binary classification. Hence, the multi-class data is converted to binary-class data. Label encoding is applied to the class attribute to convert it to numerical format.

Intrusion detection data is imbalanced. Records representing normal traffic are common and are contained in large numbers. However, records representing intrusion traffic are sparse in nature. They are considered to be rare occurrences. This nature of data tends to reduce the data quality, resulting in lowered prediction performance. Hence, this work uses an oversampling technique to balance the data and improve its quality. The first process is to identify the number of records to be generated for the data to be balanced. Every new generated record is obtained using a combination of two existing records. Although the training data is balanced, oversampling generally results in data over training due to the near-duplicated records. The proposed model handles this issue by using the sampling approach.

2.3 Feature Selection

Network data contains features depicting the type of traffic and the network through which the packet traverses. This results in a large number of features, eventually leading to the curse of dimensionality. The process of oversampling during the balancing phase also results in an increased number of instances. These processes result in an overall increase in the training data. Hence, in order to reduce the time of computation, this work integrates a feature selection approach. A tree based feature selection model is used for this purpose. The created model is a meta transformer that uses the entropy values to identify the significance of features. Decision tree algorithm is used to identify the feature importance and the features exhibiting low importance levels are eliminated from the training data. This process results in a reduction of size in the training data, hence reduced computational complexity.

2.4 Semi supervised First Level Prediction

The first level of semi-supervised prediction uses multiple heterogeneous models for the training process. A combination of supervised and unsupervised models has been used to build the first-level prediction architecture. The training data is divided into overlapping subsets to create multiple training data subsets for the machine learning models. This work uses a combination of the Gaussian Mixture and Decision Tree models as the machine learning models of choice. The Gaussian mixture is an unsupervised clustering model that assumes the input data is in Gaussian distribution. The model groups data points that belong to a single distribution into a cluster. They are probabilistic models that use soft clustering approach to distribute points into clusters. The major advantage of using the machine mixture model is that it considers the variance level in points today to determine the clusters. Hence, Gaussian mixture models can provide the probability levels of a point belonging to a cluster.

A decision tree is a tree-based modeling technique that creates branches based on the entropy levels obtained from the training data. Each tree node represents a condition and each branch represents a possible decision. The leaf nodes represent the final prediction. Although, the decision tree is a weak learning model, it can handle dynamic data ensuring effective predictions even in a streaming context.

Multiple instances of each model are created, and each model is provided with a different subset of the training data. Every model is made to train on a different subset of the actual training data. This reduces over fitting caused by oversampling. After the completion of the training process, the training data is used to identify the first level predictions. Results obtained from these predictions are integrated to form the level 2 training data for the second level stacking model. The class label is applied to this data and passed to the next level.

2.5 Final Prediction using Second Level Stacking

The second level stacked model is the meta model that uses the previous level predictions for its training process. Since this model uses previous predictions and not the training data, the model is considered to be more robust to issues like noise that are generally present in real time data. Logistic regression is used as the meta model of choice.

The logistic regression statistical analysis technique net predicts binary outcomes for the training data. The prediction is performed by analyzing the relationships between one or more existing independent variables. It estimates the parameters of the logistic model, fitting the model into a curve. Since the second level model is required to operate turn the predictions rather than the training data, logistic regression is used as a model of choice. The prediction data open from the previous level is used for training the Logistic regression model. The test data is passed to the level one models and the predictions from the semi supervised models are integrated and passed to the logistic regression model. Predictions from the logistic regression model are considered the final predictions.

3 Results and Discussion

The proposed Stacking and Feature engineering based Semi supervised (SFS) model has been implemented using Python. The SFS model has been analyzed using the KDD CUP 99 dataset, NSL- KDD dataset, and UNSW- NB15 dataset. Each data set exhibits varied imbalance levels and noise levels.

The PR plot representing precision and recall levels of the SFS model on all the 3 datasets is shown in Figure 1. High levels of precision and recall represent highly effective classifier models. The plot shows precision levels almost nearing one and recall levels greater than 90%. This shows that the model is highly capable of predicting varied datasets with high production efficiency and is generic in nature.

A comparison of the aggregate measures accuracy, F- measure and AUC is shown in Figure 2 . All three metrics are computed by combining the existing performance metrics, and they represent an overall performance level that can be used to identify the performance of the model as a whole in predicting over the binary class data. The chart shows > 90% accuracy levels on all the three datasets, greater than 90% F- measure levels and also greater than 90% AUC levels. This performance indicates at the model is capable of providing highly effective overall prediction and is not biased. The unbiased nature of the model shows that the model is not affected by the imbalance levels contained in the data.

3.1 Comparative Study

The SFS model has been compared with the SAVER-DNN [16] model. Analysis based on the ROC plot over the NSL-KDD and UNSW-NB15 datasets is shown in Figures 3 and 4. The ROC plot of NSL KDD data shows that the SFS model exhibits the highest true positive and lowest false positive levels. High true positive levels and low positive levels depict an ideal classifier model. Comparing the SFS model with the SAVER-DNN model, the SFS model exhibits a higher true positive rate reaching

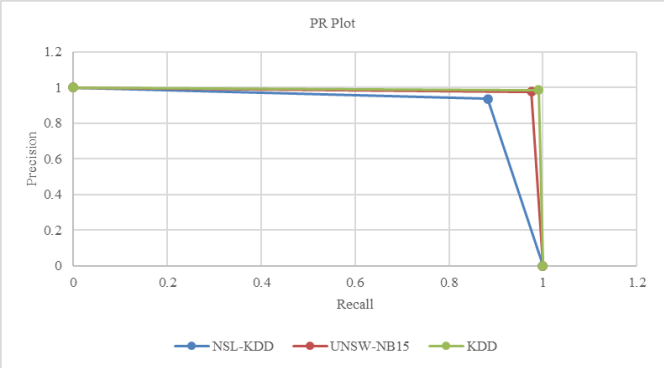

Fig 1. PR plot of SFS

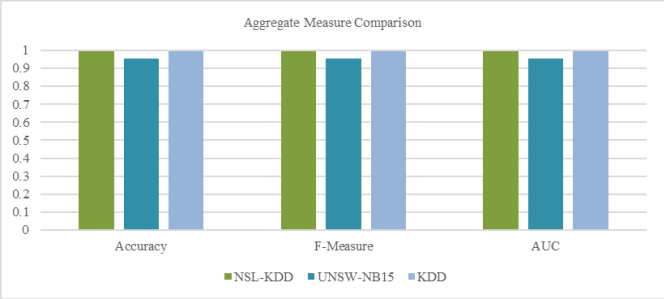

Fig 2. Aggregate Measures of SFS

Table 1. Performance Measures of SFS

Technique	NSL-KDD	UNSW-NB15	KDD
FPR	0.0004	0.0815	0.0008
TPR	0.9884	0.9929	0.9884
Recall	0.9884	0.9929	0.9884
Precision	0.9996	0.9185	0.9992
TNR	0.9996	0.9185	0.9992
FNR	0.0116	0.0071	0.0116
Accuracy	0.9936	0.9542	0.9935
F-Measure	0.9940	0.9542	0.9938
AUC	0.9940	0.9557	0.9938

almost one, while SAVER-DNN exhibits slightly reduced true positive levels. However, when considering the false positive rate, SFS exhibits almost zero false positive levels, while the false positive levels of SAVER-DNN exhibit a much higher value depicting that the model produces more false alarms.

The ROC plot on UNSW-NB15 dataset is shown inFigure 4. Both the models where observed to exhibit a certain false positive rate. Considering the true positive rate, the SFS model exhibits higher true positive levels compared to the SAVAER-DNN model.

Tabulated view of the performance is provided in Tables 2 and 3. The best predictions are highlighted in bold. Except for the FPR levels, the SFS model exhibits better predictions in all the other metrics on the UNSW-NB15 dataset. A slight reduction of 3% in FPR levels has been observed, an 8% increase in TPR levels, 2% increase in accuracy levels and 2% increase in F-Measure levels has been observed.

An analysis on NSL-KDD dataset (Table 3) indicates that the SFS model exhibits better prediction on all the metrics. A 4% reduction in FPR levels, 3% increase in TPR levels, 10% increase in accuracy levels and 9% increase in F- measure levels Indicates the high efficiency of the prediction from the SFS model.

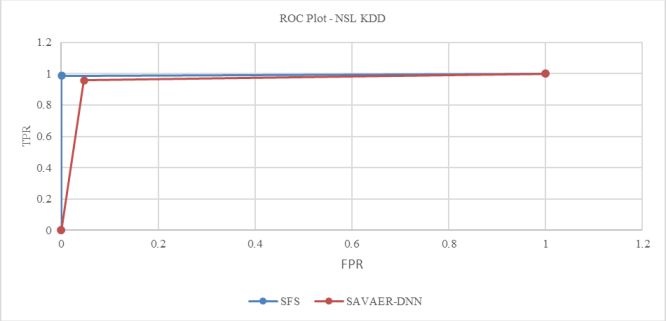

Fig 3. ROC Comparison of SFS on NSL-KDD

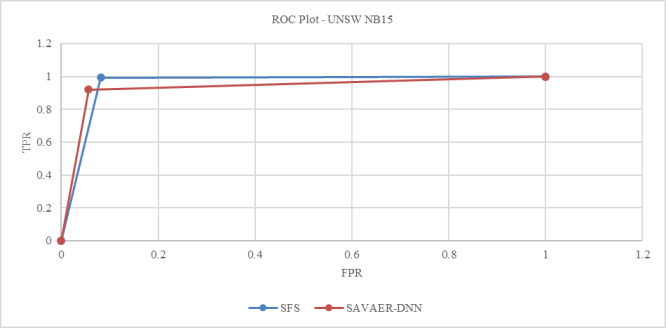

Fig 4. ROC Comparison of SFS on UNSW-NB15

Table 2. Performance Comparison of SFS on UNSW-NB15 Data

	SAVAER-DNN	SFS
FPR	0.056	0.08
TPR	0.919	0.99
Accuracy	0.930	0.95
F-Measure	0.935	0.95

Table 3. Performance Comparison of SFS on NSL-KDD Data

	SAVAER-DNN	SFS
FPR	0.047	0.00042
TPR	0.959	0.98838
Accuracy	0.89	0.99365
F-Measure	0.9	0.99397

Comparative analysis over the UNSW-NB15 dataset indicates reduced FPR levels in the SFS model. The reduction levels were observed to be 3% compared to SAVAER-DNN model. Low FPR indicated reduced false alarms. This shows that the model has very low probability of labeling a normal transaction (majority class) as an anomalous record (minority class). Improved TPR levels at 8% by SFS indicates that the model can more effectively distinguish a minority class record from a majority class record compared to the SAVAER-DNN model. This indicates that the SFS model is much more capable of handling data imbalance compared to SAVAER-DNN. The accuracy level has been observed to be 95%, which is a 2% increase from SAVAER-DNN indicates better overall performance. This indicates that the model is capable of accurately classifying both minority and majority classes, showing the high efficiency of SFS.

Comparative analysis over the NSL-KDD data indicates ~0% FPR levels by the SFS model, indicating almost no false alarm levels. Further, the TPR levels were recorded to be 98%, which is 3% more than the SAVAER-DNN model. These metrics indicate better imbalance handling and also qualitative performance. Performance on KDD CUP 99 data indicates high performance

with >98% performance on all the metrics.

Overall analysis based on the KDD CUP, NSL-KDD and UNSW-NB15 datasets indicate that the model is highly capable of providing high quality performance over varied datasets with different data distributions and varied imbalance levels. This generic nature of the model elucidates that the model can effectively perform in real-time data from the network.

4 Conclusion

This work presents an intrusion detection architecture that integrates a data balancing module and also a prediction module that can handle the imbalance to improve the detection process. The proposed Stacking and Feature engineering-based Semi supervised (SFS) model uses oversampling technique to balance the data and a stacking architecture that integrates Supervised and Semi-supervised modeling techniques for prediction. The issue of over training introduced due to oversampling is handled by the stacking architecture. Experimental results indicate high performance of 95% accuracy on UNSW-NB15 dataset, 99% accuracy on NSL-KDD dataset and 99% accuracy on KDD CUP 99 dataset. Each dataset is composed of varied data distributions and imbalance levels. The novelty of this work lies in the architecture which includes components that handle additional issues like data imbalance and high data complexity. Further, the domain centric nature of the model results in providing a generic architecture that can be adopted to varied data distributions. However, the SFS model exhibits slightly increased false alarm levels on UNSW-NB15 dataset. This is considered to be due to the highly imbalanced nature of data. Future enhancements will deal on proposing an architecture that identifies the imbalance levels and selects models accordingly.

References

1) Gui L, Yuan W, Xiao F. CSI-based passive intrusion detection bound estimation in indoor NLoS scenario. *Fundamental Research*. 2022. Available from: https://doi.org/10.1016/j.fmre.2022.05.015.
2) T Y, Murtugudde G. An efficient algorithm for anomaly intrusion detection in a network. *Global Transitions Proceedings*. 2021;2(2):255–260. Available from: https://doi.org/10.1016/j.gltp.2021.08.066.
3) Baldini G, Amerini I. Online Distributed Denial of Service (DDoS) intrusion detection based on adaptive sliding window and morphological fractal dimension. *Computer Networks*. 2022;210:108923. Available from: https://doi.org/10.1016/j.comnet.2022.108923.
4) Fu Y, Du Y, Cao Z, Li Q, Xiang W. A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. *Electronics*. 2022;11(6):898. Available from: https://doi.org/10.3390/electronics11060898.
5) Rani M, Gagandeep. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. *Multimedia Tools and Applications*. 2022;81:8499–8518. Available from: https://doi.org/10.1007/s11042-021-11747-6.
6) Pimsarn C, Boongoen T, Iam-On N, Naik N, Yang L. Strengthening intrusion detection system for adversarial attacks: improved handling of imbalance classification problem. *Complex & Intelligent Systems*. 2022;8:4863–4880. Available from: https://doi.org/10.1007/s40747-022-00739-0.
7) Guarascio M, Cassavia N, Pisani FS, Manco G. Boosting Cyber-Threat Intelligence via Collaborative Intrusion Detection. *Future Generation Computer Systems*. 2022;135:30–43. Available from: https://doi.org/10.1016/j.future.2022.04.028.
8) Jordan B, Piazza R, Darley T. 2020. Available from: https://docs.oasis-open.org/cti/stix/v2.1/stix-v2.1.html.
9) Le TTH, Oktian YE, Kim H. XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems. *Sustainability*. 2022;14(14):8707. Available from: https://doi.org/10.3390/su14148707.
10) Darley T, Kirillov I, Piazza R, Beck D. Taxii^{TM}version 2.1 committee specification 01. 2020.
11) Prasad R, Shankar S. secure intrusion detection system routing protocol for mobile ad-hoc network. *Global Transitions Proceedings*. 2022;4(4):1–11. Available from: https://doi.org/10.1016/j.gltp.2021.10.003.
12) Wu T, Fan H, Zhu H, You C, Zhou H, Huang X. Intrusion detection system combined enhanced random forest with SMOTE algorithm. *EURASIP Journal on Advances in Signal Processing*. 2022;2022(1). Available from: https://doi.org/10.1186/s13634-022-00871-6.
13) Siddiqui F, Beley J, Zeadally S, Braught G. Secure and lightweight communication in heterogeneous IoT environments. *Internet of Things*. 2021;14:100093. Available from: https://doi.org/10.1016/j.iot.2019.100093.
14) Mehmood A, Khanan A, Umar MM, Abdullah S, Ariffin KAZ, Song H. Secure Knowledge and Cluster-Based Intrusion Detection Mechanism for Smart Wireless Sensor Networks. *IEEE Access*. 2018;6:5688–5694. Available from: https://doi.org/10.1109/ACCESS.2017.2770020.
15) Yang Y, Zheng K, Wu B, Yang Y, Wang X. Network Intrusion Detection Based on Supervised Adversarial Variational Auto-Encoder With Regularization. *IEEE Access*. 2020;8:42169–42184. Available from: https://doi.org/10.1109/ACCESS.2020.2977007.
16) Asif M, Abbas S, Khan MA, Fatima A, Khan MA, Lee SW. MapReduce based intelligent model for intrusion detection using machine learning technique. *Journal of King Saud University - Computer and Information Sciences*. 2021. Available from: https://doi.org/10.1016/j.jksuci.2021.12.008.
17) Yao R, Wang N, Liu Z, Chen P, Ma D, Sheng X. Intrusion detection system in the Smart Distribution Network: A feature engineering based AE-LightGBM approach. *Energy Reports*. 2021;7:353–361. Available from: https://doi.org/10.1016/j.egyr.2021.10.024.
18) Priya AS, Ramesh SB, Kumar. Intrusion Detection using Attribute Subset Selector Bagging (ASUB) to Handle Imbalance and Noise". *International Journal of Computer Science and Network Security*. 2022;22:97–102. Available from: https://doi.org/10.22937/IJCSNS.2022.22.5.15.
19) Parashar, Saggu, Garg. Machine learning based framework for network intrusion detection system using stacking ensemble technique. *Indian Journal of Engineering and Materials Sciences*. 2022;29(04). Available from: https://doi.org/10.56042/ijems.v29i4.46838.