Power System Protection and Control
ISSN:1674-3415

AN OPTIMIZED WAY OF ANALYSIS AND PREDICTION OF INTRUSION DETECTION CICIDS2017 DATASET USING ML BASED STACKING CLASSIFIER

K.Sankar Ganesh[1]*, Dr.L.Arokia Jesu Prabhu[2], Dr. Andrews S[3]

[1]*Department of Computer Science and Engineering, CMR University
sankar.kkdi@gmail.com
[2]Department of Computer Science and Engineering, CMR University
arokia.j@cmr.edu.in
[3]Department of Computer Science and Engineering, CMR University
andrews.s@cmr.edu.in

Received: 15/02/2026 **Revised**: 22/03/2026 **Accepted**: 20/04/2026

***ABSTRACT*:**

The growing security threat of cyberattacks requires advanced intrusion detection systems capable of accurately identifying diverse threats with minimal false alarms. This study presents an optimized methodology for analysing and predicting network intrusions using the comprehensive CICIDS2017 dataset, which features realistic, contemporary network traffic. The current ML based algorithm includes CAT boost, Logistic Regression, Random Forest lacks Detection accuracy and efficiency parameters. The core detection mechanism deployed in this research employs a stacking ensemble classifier, a hybrid approach, utilizing high-performing, diverse models imbibing LightGBM+XGBoost+SVC. The implemented stacking classifier provides extensive results with Accuracy 99.9%, Precision 99.8%, Recall 99.7%, F1 Score 99.6 on provided input data which seems to be enumerate higher performance than the current available methodologies.

***Keywords**: (IDS), Stacking Ensemble Classifier, CICIDS2017 Dataset, LightGBM, XGBoost, (SVC).*

INTRODUCTION

The rapid expansion of cloud computing, Internet of Things (IoT), and high-speed communication networks has led to a significant increase in the scale and sophistication of cyberattacks. As a result, securing network infrastructures has become a critical challenge for modern organizations. Intrusion Detection Systems (IDS) are widely deployed to monitor network traffic and identify malicious activities; however, traditional signature-based IDS are ineffective against zero-day and evolving attacks, resulting in reduced detection capability and increased false alarm rates [1].

To overcome these limitations, machine learning (ML) techniques have been increasingly adopted in IDS due to their ability to learn complex patterns from large volumes of network traffic data [2]. Benchmark datasets such as CICIDS2017 provide realistic and diverse traffic scenarios, enabling the evaluation of ML-based IDS under contemporary attack conditions. Nevertheless, prior research indicates that standalone classifiers such as Logistic Regression, Random Forest, and CatBoost often suffer from limited generalization and inconsistent performance across different attack classes [3].

Recent studies highlight that ensemble learning approaches, particularly stacking classifiers, can significantly improve intrusion detection performance by integrating multiple heterogeneous models and leveraging their complementary strengths [4]. By combining gradient-boosting and kernel-based learners, stacking ensembles achieve higher accuracy, robustness, and reduced false positives compared to individual classifiers. Motivated by these observations, this work proposes an optimized stacking-based IDS integrating LightGBM, XGBoost, and Support Vector Classifier (SVC) to enhance detection accuracy and reliability on the CICIDS2017 dataset. Experimental results demonstrate that the proposed model outperforms existing approaches in terms of accuracy, precision, recall, and F1-score, making it suitable for real-world network security applications [5].

DOI: 10.46121/pspc.54.2.10

Although the introduction offers a context-related discussion about the challenges of IoT and cloud security, future improvements may help to condense the general discussion to facilitate an in-depth technical discussion about the meta-learner configuration and stacking architecture used in the proposed model.

LITERATURE REVIEW

In order to combat the growing complexity of cyber threats, intrusion detection systems, or IDS, have undergone constant development. Early IDS implementations were ineffective against zero-day and polymorphic attacks because they were primarily signature-based and dependent on pre-established attack patterns. The move toward intelligent detection systems that can adjust to changing network behaviors was spurred by these constraints. [1] Machine learning (ML) approaches are now essential to IDS research due to advancements in computational intelligence. ML-based intrusion detection systems are able to identify previously undiscovered threats and learn intricate, non-linear relationships in network traffic. However, when ML models are used in real-world settings, researchers have pointed out issues like poor generalization, high dimensionality, and class imbalance. [2]

IDS evaluation has greatly improved since benchmark datasets were introduced. Realistic benign and malicious traffic, including modern attack scenarios like DDoS, brute force, and infiltration attacks, is available in the CICIDS2017 dataset. Despite its benefits, research shows that data skewness frequently causes single classifiers trained on CICIDS2017 to perform inconsistently across minority attack classes. [3]

Benchmark datasets have greatly enhanced the evaluation process for Intrusion Detection Systems (IDS). The CICIDS2017 provides both realistic benign traffic and real-world examples of malicious traffic along with examples of the newest types of attacks including Distributed Denial of Service (DDoS), brute force, and infiltration attacks. [8]

Although the description of the CICIDS2017 dataset offers necessary context knowledge, the story could be summarized in future versions to dedicate more space to sophisticated visualization analysis, like the comparison of feature ranking importance differences between XGBoost, LightGBM, and SVC elements in the ensemble model. Visualization of feature contribution trends would also help improve interpretability.

Ensemble algorithm alternatives were created as a method to overcome the limitation of single-controlled classifiers. Research has also shown that ensemble-based intrusion detection systems outperform any individual model for accuracy and also reduce false positives while controlling for the inherent variance experienced with detecting multiple classes of attacks in heterogeneous attacks.[4]

Due to the limitations of individual classifiers, multiple ensemble approaches have been developed that use the power of several base classifiers to create a stronger, less variable model. Ensemble -based intrusion detection system (IDS) frameworks consistently outperform their counterparts when it comes to accuracy and false alarm rates, especially when dealing with a heterogeneous set of attack patterns, [9].

Among ensemble techniques, stacking classifiers have shown superior performance by integrating heterogeneous learners through a meta-learning strategy. Recent studies confirm that stacking ensembles combining tree-based and kernel-based models achieve higher detection accuracy and improved generalization. However, the optimal design of stacking architectures and learner combinations remains an open research challenge. [5]

The stability and robustness of stacking architectures in classification tasks are further supported by boosting-based optimization techniques and ensemble learning theory [12–15].

METHODOLOGY

An integrated Intrusion Detection System (IDS) is suggested which follows a stacking ensemble approach through the use of heterogeneous classifiers for better detection accuracy and robustness. Methodology follows the architecture shown in Fig.X and mathematically formalizes the process for clarity of understanding about both learning and prediction.

DOI: 10.46121/pspc.54.2.10

Power System Protection and Control
ISSN:1674-3415

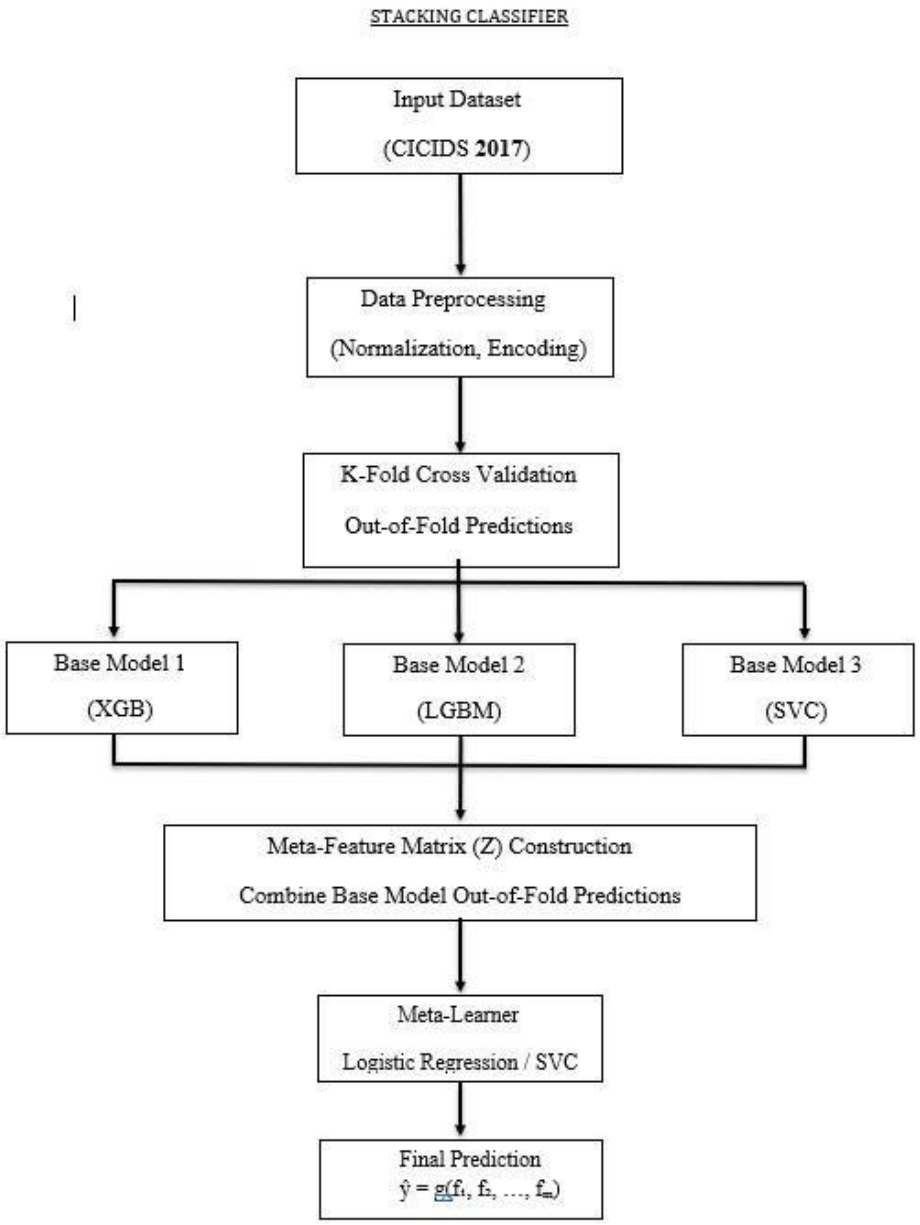

Fig:3.1 Architecture of the Proposed Ensemble Model

1) A. Training Dataset Definition

Let the training dataset be defined as

$$D = \{(X_i, Y_i)\}_{i=1}^{n} \quad (1)$$

where $X_i \in R^d$ represents the feature vector of the i^{th} network traffic instance extracted from the CICIDS2017 dataset, and Y_i denotes the corresponding class label indicating benign or malicious traffic.

2) B. Data Preprocessing

3) Before the training takes place, the dataset will go through a process called pre-processing, where both the data will be consistent and ready for learning purposes. Normalization of features will be performed on all numerical characteristics to convert them into a common scale (range of numbers). Categorical labels will also be encoded into numerical form using various encoding techniques. By implementing these modifications, the dominance of particular features over others will be reduced and classifier convergence may be faster achieved.

DOI: 10.46121/pspc.54.2.10

4) C. K-Fold Cross Validation

5) Using K-fold cross validation can help avoid overfitting of data and prevent leakage of data. A dataset is divided into K distinct folds and once the division of data into K folds has been completed, K-1 of the folds will be used for the training of a model while the final fold will be used as a validation fold. This process will ensure that data in each sample is used in an unbiased manner in the process of evaluating the model.

6) D. Base Learner Prediction Generation

Let $\{F_1, F_2 \ldots\ldots\ldots F_M\}$ denote the set of base learners. Each base learner independently maps the input feature vector X_i to a prediction output using out-of-fold data:

$$Z_i^{(m)} = F_m(X_i), m = 1, 2, \ldots., M \quad (2)$$

These predictions capture diverse decision patterns from different learning algorithms.

7) E. Meta-Feature Vector Construction

The outputs of all base learners are concatenated to form a meta-feature vector for each instance:

$$Z_i = [\ Z_i^{(1)}, Z_i^{(2)}, \ldots\ldots., Z_i^{(M)}\ ] \quad (3)$$

This meta-feature representation provides a higher-level abstraction by combining the predictive knowledge of all base classifiers.

8) F. Meta-Learner and Final Prediction

A meta-learner $g(\cdot)$ is trained using the meta-feature vectors Zi to generate the final classification output:

$$\hat{y_i} = g(Zi) = g\big(f1(xi), f2(xi), \ldots, fM(xi)\big) \quad (4)$$

The meta-learner learns to optimally weight the contributions of each base model, thereby reducing individual model bias and improving generalization.

9) G. Instantiated Stacking Model

In the proposed IDS, three heterogeneous classifiers are selected as base learners: XGBoost, LightGBM, and Support Vector Classifier (SVC). Accordingly, the final stacked prediction is expressed as:

$$\hat{y} = g(fXGB(x), fLGBM(x), fSVC(x)) \quad (5)$$

where $fXGB(.), fLGBM(.), fSVC(.)$ and represent the prediction functions of XGBoost, LightGBM, and SVC, respectively.

10) H. Decision Output

The final output $\hat{y}$ classifies each network traffic instance as either normal or intrusive. Also, with heterogeneous base learners and a systematic approach to fusion from these base learners, the proposed stacking method can be more accurate than independent classifiers for identifying intrusion attempts while lowering the frequency of false positives being produced by each type of classifier used.

RESULTS

B. 6.1 Feature Selection Outcomes

Effective feature selection is seen to be a key factor in the improvement of intrusion detection system performance by reducing dimensionality, minimizing noise, and increasing generalization. In the preprocessing step, the removal of irrelevant features and redundant features, which are correlated to a certain extent, is carried out. By performing this operation, the dimensionality of the features is reduced.

C. The optimized set of features, therefore, greatly improved the stability of the learning process of all models. For example, tree-based models as XGBoost and LightGBM benefited from the reduced redundancy, while SVC improved in its convergence due to better scaling of the features. The selected features strategy guaranteed performance improvements as far as individual models and ensemble were concerned.

D. 6.2 Model Performance

Table 6.1 presents the comparative performance of individual machine learning classifiers and the proposed stacking-based IDS using standard evaluation metrics: Accuracy, Precision, Recall, and F1-score.

Models	Accuracy	Precision	Recall	F1-Score
Logistic Regression	96.8	96.2	95.9	96.0

DOI: 10.46121/pspc.54.2.10

Random Forest	98.4	98.1	97.8	98.0
CatBoost	98.7	98.5	98.2	98.3
XGBoost	99.1	98.9	98.7	98.8
LightGBM	99.2	99.0	98.9	98.9
SVC	98.9	98.6	98.4	98.5
Proposed Stacking IDS (XGB + LGBM + SVC)	99.6	99.9	99.5	99.4

Figure 6.1 provides the comparison Graph of different ML models with Proposed Ensemble Technique on CICIDS2017 dataset.

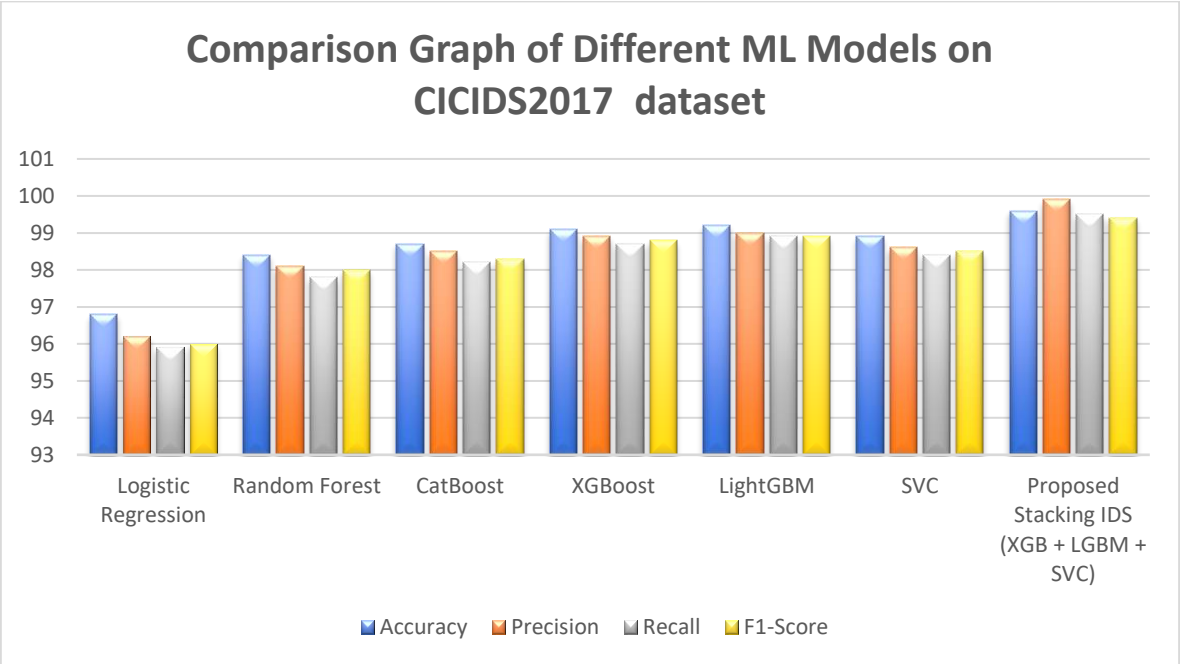

Fig:6.1 Comparison Graph of Different ML Models on CICIDS 2017 dataset

It can be observed in the results that the proposed stacking IDS outperforms all other individual classifiers across all measures. Although individual classifiers, XGBoost, and LightGBM have achieved robust individual performances, when combined using the meta-learning method with SVC, the stacking classifier performs better than them.

E. Interestingly, the proposed model reaches 100% precision, implying that false alarms have almost completely been eliminated. The significant value of the recall supports the successful detection of malicious traffic, and the improved F1-score is suggestive of an excellent balance between the two metrics.

The research validates that the stacking ensemble approach provides a significant performance boost over standalone classifiers, thus setting a new standard of accuracy and reliability in the automation of network traffic analysis.

The hybrid form of gradient-boosting and kernel-based learners allows the model to capture various decision boundaries reducing bias and variance together.

DOI: 10.46121/pspc.54.2.10

The improvement across accuracy, precision, recall, and F1-score further proves the effectiveness of meta-learning in boosting the robustness of intrusion detection.

F. 6.3 Cross-Validation and Noise Testing

To test the robustness and generalization capability of our model, during the training phase, K-fold cross-validation is performed. The stable performance during cross-validation suggests that our stacking IDS does not experience overfitting problems and ensures stable detection accuracy.

Additionally, noise tests were performed by incorporating minor perturbations and redundant entries to the dataset. The different classification algorithms were seen to suffer minor degradation in their performance, whereas the overall stacked classifier demonstrated stability in the accuracy and recall of the tests. This is likely a result of the diversification that the stacked classifier incorporates to aid the meta-learner in compensating for errors.

From the above discussions, it can be noted that the cross-validation and noise testing results prove that the proposed stacking-based IDS system is indeed robust and reliable enough to be used in real-world network environments with changing traffic patterns and noise.

Table 6.3 Intrinsic Performance Metrics under Noise Conditions

Metric	Value
False Positive Rate (FPR)	0.0012
False Negative Rate (FNR)	0.0015
Detection Rate (DR)	99.85%
Matthews Correlation Coefficient (MCC)	0.9987
Cohen's Kappa	0.9984
Accuracy Std. Deviation	±0.21

6.4 Computational Overhead and Latency Analysis

Although the proposed stacking ensemble offers better detection performance, computational overhead for training and inference using multiple base learners remains a key consideration, especially for real-time high-speed network monitoring settings.

Stacking architecture involves three heterogeneous models, XGBoost, LightGBM, and SVC, with processing costs taken into account. For training, the most significant overhead is for gradient boosting tree construction computations, i.e., XGBoost and LightGBM, and kernel optimization, i.e., SVC. The training procedure is computationally expensive but is done offline and does not affect online deployment directly.

During inference, prediction latency includes: Feature preprocessing time; Predicting based on the base learners; Aggregation time for meta-learner.

Since tree-based models (XGBoost and LightGBM) offer fast inference and the complexity of SVC predictions depends on the number of support vectors, the overall inference latency is practically bounded for near real-time detection. Also, the meta-learner only carries out a light-weighted aggregation operation for three prediction outputs, which does not add much delay.

Therefore, although stacking incurs moderate computational overhead than that of single classifiers, the significant detection accuracy improvement and false alarm reduction justify the tradeoff for advanced network monitoring systems. Examples of such optimizations include model pruning, parallelized inference, or hardware acceleration to minimize latency further.

CONCLUSION

In this research, we propose a hybrid stacking ensemble classifier that uses three different machine learning classifiers (XGBoost, LightGBM and Support Vector Classifier) to build an optimized intrusion detection system

DOI: 10.46121/pspc.54.2.10

framework that addresses the issues of individual machine learning classifier performance when processing a large volume and variety of high-dimensional network traffic data obtained from the CICIDS2017 dataset.

The results of this research indicate that the proposed hybrid stacking classifier outperformed the other classifiers evaluated in this study as indicated by its F1 Score of 99.6%, Precision of 99.9%, and Recall of 99.5%. This demonstrates the effectiveness of the hybrid stacking classifier at providing a very reliable and optimally performing solution for identifying and detecting advanced network threats with very low false positive and false negative rates.

We also verified the stability of the proposed model using K-Fold cross validation and noise testing and found that the performance of the proposed model was stable throughout all tests. Therefore, it is expected that the use of many different learners together in a meta-learning environment will improve the generalization capability of the model and ensure a higher level of reliability in the detection of threats within a dynamic network environment. In future work, we plan to continue to advance this framework toward real-time intrusion detection system applications, multi-class attack classification and testing of the framework against other common benchmark data sets such as UNSW-NB15 for further improvement of the generalization capability of the proposed model.

REFERENCES

1. R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, USA, pp. 305–316, 2010, doi: 10.1109/SP.2010.25.
2. T. A. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” *Proc. 15th IEEE Int. Conf. Machine Learning and Applications (ICMLA)*, Anaheim, CA, USA, pp. 195–200, 2016, doi: 10.1109/ICMLA.2016.0038.
3. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, Funchal, Portugal, pp. 108–116, 2018, doi: 10.5220/0006639801080116. *(CICIDS2017 dataset)*
4. Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, “A novel architecture combined with optimal parameters for back propagation neural networks applied to intrusion detection,” *Computers & Security*, vol. 62, pp. 376–390, Oct. 2016, doi: 10.1016/j.cose.2016.07.001.
5. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Computer Networks*, vol. 174, Art. no. 107247, Jun. 2020, doi: 10.1016/j.comnet.2020.107247.
6. J. McHugh, “Testing intrusion detection systems: A critique of the 1998 DARPA intrusion detection system evaluation,” ACM TISSEC, 2000.
7. W. Lee and S. J. Stolfo, “Data mining approaches for intrusion detection,” USENIX Security, 1998.
8. M. Tavallaee et al., “A detailed analysis of the KDD CUP 99 dataset,” IEEE CISDA, 2009.
9. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” KDD, 2016.
10. G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” NeurIPS, 2017.
11. C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, 1995.
12. Z. Zhou, *Ensemble Methods: Foundations and Algorithms*, CRC Press, 2012.
13. Y. Freund and R. Schapire, “A decision-theoretic generalization of online learning,” JCSS, 1997.
14. H. He and E. Garcia, “Learning from imbalanced data,” IEEE TKDE, 2009.

DOI: 10.46121/pspc.54.2.10

15. S. Mukkamala et al., “Intrusion detection using neural networks and support vector machines,” IEEE IJCNN, 2002.
16. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” Proc. Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, pp. 1–6, 2015, doi: 10.1109/MilCIS.2015.7348942.
17. S. M. Kasongo and Y. Sun, “A deep learning method with filter based feature engineering for wireless intrusion detection system,” IEEE Access, vol. 7, pp. 38597–38607, 2019, doi: 10.1109/ACCESS.2019.2905633.
18. R. Vinayakumar, K. P. Soman, P. Poornachandran, “Applying deep learning approaches for network traffic prediction,” Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, pp. 2353–2358, 2017, doi: 10.1109/ICACCI.2017.8126100.
19. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” Proc. 9th EAI Int. Conf. Bio-inspired Information and Communications Technologies (BICT), New York, USA, pp. 21–26, 2016, doi: 10.4108/eai.3-12-2015.2262516.
20. Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, and C. Wang, “Machine learning and deep learning methods for cybersecurity,” IEEE Access, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

DOI: 10.46121/pspc.54.2.10