Revolutionizing Wireless Network Intrusion Detection: Leveraging Adaptive Synthetic Sampling and Enhanced Convolutional Neural Networks for Improved Cybersecurity

J. Prashanthi[1*], Sai Sudha Lahari[1], G. Sai Sathwika[1], Lokesh[1], Rakesh[1]

[1]Department of Computer Science and Engineering, Sree Dattha Group of Institutions, Hyderabad, Telangana, India

[*]Corresponding E-mail: prashanthi@sreedattha.ac.in

ABSTRACT

In recent years, the increasing reliance on wireless networks has made them a prime target for cyberattacks. Intrusion detection systems (IDS) are critical for identifying and mitigating such threats. Traditional IDS methods have limitations, and leveraging advanced techniques like deep learning and synthetic data generation can enhance the effectiveness of intrusion detection. Traditional wireless network intrusion detection systems often rely on rule-based algorithms or signature-based detection methods. While effective to some extent, these methods may struggle with detecting novel or evolving attack patterns. The primary challenge is to design a wireless network intrusion detection system that can accurately identify and respond to various types of cyberattacks. This involves developing a model capable of learning complex patterns indicative of intrusions while adapting to changing attack strategies. Therefore, the need of wireless networks become more prevalent, the need for robust and accurate intrusion detection systems is paramount. Rapidly evolving cyber threats require adaptive and sophisticated approaches to safeguard network integrity and data confidentiality. The project addresses this need by combining advanced techniques for more effective intrusion detection. This aims to revolutionize wireless network security by combining two powerful techniques. The first is adaptive synthetic sampling, which generates additional training data to balance class distributions and improve model performance. The second is an enhanced convolutional neural network (CNN), which is adept at learning complex spatial patterns in network traffic data. By integrating these approaches, this research endeavors to develop a system capable of accurately detecting a wide range of intrusions in wireless networks. This advancement holds great promise for significantly enhancing the security posture of wireless networks, protecting critical assets and data from cyber threats.

Keywords: Wireless Networks, Intrusion Detection System, Data Balancing, Convolutional Neural Network.

1. Introduction

The rapid adoption of wireless networks in India has been fuelled by the increasing penetration of internet services and mobile devices. These networks, including Wi-Fi and mobile networks, have become essential for personal and professional communication, making them prime targets for cyberattacks. As India transitions towards a more digital economy, securing these networks has become a critical concern. Traditional IDS, which often rely on rule-based or signature-based methods, face significant limitations in detecting novel and evolving cyber threats. The history of wireless network security in India mirrors the global evolution of cybersecurity measures. In the early 2000s, security practices were rudimentary, focusing on basic encryption and password protection. The primary security protocols in use were WEP (Wired Equivalent Privacy) and later WPA (Wi-Fi Protected Access). With the rise of mobile internet usage and smart devices in the 2010s, there was a shift towards more robust security measures, including WPA2 and the introduction of IDS and Intrusion Prevention Systems (IPS). The 2020s have seen the emergence of advanced persistent threats (APTs), ransomware, and sophisticated malware, prompting the integration of machine learning and artificial intelligence into cybersecurity frameworks. This period also marked an increased emphasis on real-time threat detection and response.

India is home to over 850 million internet users as of 2023, making it the second-largest online market globally. Additionally, the country has approximately 1.17 billion mobile subscribers, with a significant portion using mobile data services. The number of public Wi-Fi hotspots has been increasing, with a target of establishing 10 million Wi-Fi hotspots by 2022 under the National Broadband Mission.

Cybercrime in India has seen a significant increase, with a reported 18.4% rise in 2021 compared to the previous year. This includes phishing attacks, ransomware, and data breaches. The financial impact of these cyberattacks is substantial, with businesses facing an average cost of $3.86 million per breach as of 2022. Key sectors such as banking, healthcare, and government services are frequent targets of cyberattacks, necessitating robust security measures. To combat these threats, the Indian government has launched several initiatives. The National Cyber Security Policy (2013) aims to create a secure cyber ecosystem in the country, focusing on enhancing the security of information infrastructure. CERT-In (Indian Computer Emergency Response Team) serves as the nodal agency for responding to cybersecurity incidents, playing a crucial role in coordinating responses to cyber threats. Additionally, the proposed Data Protection Bill aims to enhance data privacy and security for individuals and organizations in India.

Challenges and Need

Despite advancements in cybersecurity measures, traditional IDS face significant challenges in detecting sophisticated and evolving threats. These challenges include the difficulty in detecting novel and unknown attack patterns, the high rate of false positives and negatives, and scalability issues due to the increasing volume of network traffic. Traditional IDS often generate false alerts, leading to alert fatigue and potentially missing actual threats. The need for robust and accurate intrusion detection systems in India's wireless networks is paramount. Advanced techniques such as adaptive synthetic sampling and enhanced convolutional neural networks offer promising solutions to these challenges. Adaptive synthetic sampling generates additional training data to balance class distributions and improve model performance, while enhanced convolutional neural networks are adept at learning complex spatial patterns in network traffic data. By addressing the limitations of traditional methods and leveraging these advanced techniques, India can significantly enhance the security of its wireless networks, protecting critical assets and data from an ever-evolving landscape of cyber threats.

2. Literature Survey

The development of attack recognition technology has gone through three stages: pattern matching algorithms, machine learning algorithms and deep learning algorithms. The pattern matching algorithm was first applied to intrusion detection tasks based on feature matching. In [1], Wu and Shen analyzed the classical pattern matching algorithms, BM and AC, and proposed the corresponding improved algorithms BMHS and AC-BM; experiments illustrated that the enhanced algorithms greatly optimize the timeliness of IDS. Dagar et al. [2] applied RabinKarp and Knuth-MorrisPratt pattern matching algorithms to intrusion detection tasks and compared their execution efficiency. However, these pattern matching algorithms are difficult to adapt to today’s network environment due to the diversity of network attacks. An attack recognition algorithm based on ML has been successfully applied to IDS and achieved excellent performance, which gradually replaced the traditional pattern matching algorithm. SVM is a typical supervised learning model in ML.

Thaseen and Kumar [3] presented a novel attack recognition model based on chi-square feature selection and multi class support vector machine (SVM). The simulation illustrates that removing redundant features significantly improves the calculation accuracy and execution efficiency of the model. Ingre et al. [4] established a novel intrusion recognition system by combining the relevant feature screening algorithm with decision trees. Nancy et al. [5] designed a dynamic recursive feature selection algorithm. By extending the decision tree algorithm and combining it with convolutional neural networks. They proposed an intelligent fuzzy temporal decision tree algorithm. The new algorithm achieved a high detection rate of unknown attacks on the KDD cup dataset. An improved IDS based on a Bayesian network and feature selection algorithm was proposed in [6]. Although these ML detection algorithms achieved higher recognition accuracy in intrusion detection tasks, they not only need large-scale feature engineering but the model parameters are also difficult to adjust. However, the DL algorithm can autonomously abstract features from basic network traffic without complex feature engineering. Therefore, related research on intrusion detection is gradually focused on the DL method. An LSTM classifier with a gradient descent optimizer is used in IDS [7], which can effectively mine the association between features from the perspective of time.

Su et al. [8] combined an attention mechanism and BLSTM (bidirectional long short-term memory) to propose a network anomaly detection model BAT, which extracts coarse-grained features by connecting forward LSTM and backward LSTM. The BAT model uses an attention mechanism to filter the network flow vectors generated by the BLSTM model to obtain the key characteristics of network traffic classification. Wei et al. [9] applied particle swarm optimization (PSO) to optimize the structure of DBN, and the improved DBN achieves significant anomaly detection ability.Gao et al. [10] designed an effective attack recognition method by combining association rules and improved deep neural networks (DNN), which uses the apriori algorithm to mine the association between discrete features and labels to improve the recognition accuracy.

Yin et al. [10] presented an effective attack recognition model by using feature enhancement and improved RNN; however, the feature enhancement method also increases the computational complexity of the model. CNN has been successfully applied to intrusion detection tasks because it can extract network traffic characteristics more effectively [4]. Lin et al. [5] designed a character-level CL-CNN model. The character-based encoding method makes the features more discretized, which contributes to improving the detection accuracy of IDS. Wu et al. [11] designed a CNN model with a simple structure and proved the necessity of converting the original data into a 2D format through experiments. In addition, the combination of this simple CNN and 2D data conversion greatly improves the detection efficiency of the model compared with the RNN model in [12]. Ding and Zhai [5] proposed a convolutional neural network model (MS-CNN) based on multistage features. Multistage features are obtained by connecting the outputs of all convolutional layers to the dense layer with the softmax classifier. By adding supplementary information (such as local information and detailed information lost by higherlevel convolutional layers), the expressive ability of the model significantly improves. Yang and Wang [13] extracted diverse features through a cross-layer aggregated CNN model, which greatly improved the expression ability of the model. Although the above attack recognition algorithms using CNN improve the detection accuracy, they ignore the interchannel information redundancy in the convolution layer. However, we cannot directly discard some channel information because we are not sure which channels are redundant. To reasonably eliminate the problem of interchannel information redundancy, Zhang et al. [14] proposed a split-based plug and play convolution (SPC) block, which divides the channel of the convolution layer into two parts, the representative part and the uncertain redundant part, after which hierarchical processing is performed. Inspired by this idea, we propose an SPC-equipped CNN (SPC-CNN) and apply it to attack recognition tasks.

3. Proposed methodology

This represents a comprehensive approach to intrusion detection in wireless networks using a combination of preprocessing techniques, various classification algorithms, and performance evaluation metrics as shown in Figure 1.

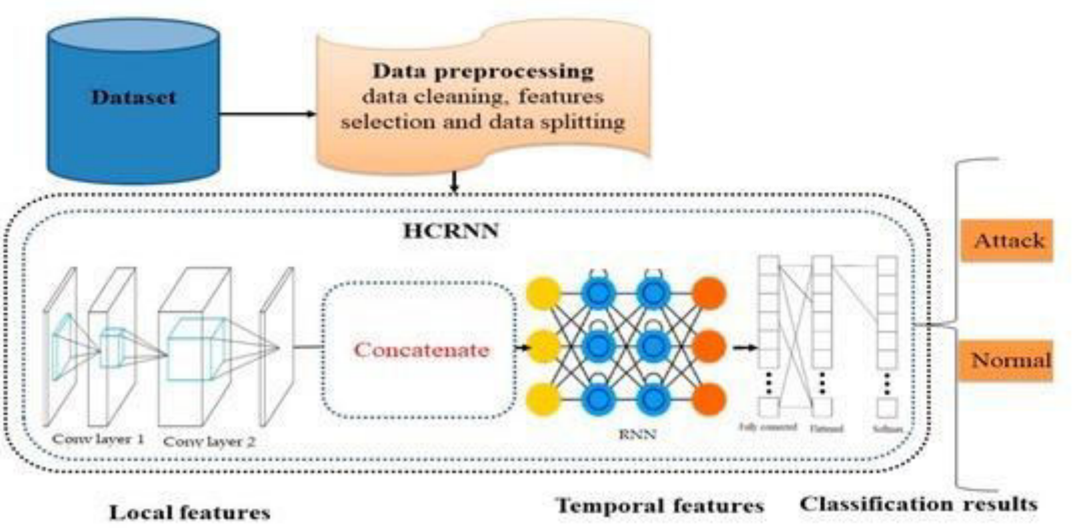


Figure 1: Proposed system architecture.

Dataset Upload and Exploration: The initial step involves uploading the NSL-KDD dataset, a common dataset for intrusion detection. The Tkinter GUI facilitates user-friendly file selection, and the dataset is displayed to provide a quick overview.

Preprocessing: After uploading the dataset, preprocessing steps are applied to ensure it is ready for model training. The process involves handling missing values, label encoding for categorical features, and normalization. The LabelEncoder from scikit-learn is utilized to convert categorical variables into numerical format, making them suitable for machine learning algorithms.

Data Augmentation using SMOTE: The code implements Synthetic Minority Over-sampling Technique (SMOTE) for data augmentation. This technique helps balance the class distribution by oversampling the minority classes. A bar graph visualizes the distribution of different attack types before and after augmentation.

Train-Test Split: The dataset is split into training and testing sets, with 80% of the records used for training and 20% for testing. This step ensures the model's performance is evaluated on unseen data.

Training a Specialized CNN Model (ECNN): The code defines and trains a specialized Convolutional Neural Network (CNN) model, referred to as ECNN (Enhanced CNN), using the Keras library. The model architecture includes convolutional layers, max-pooling layers, and dense layers. The training process involves saving the best model weights to a file for later use.

Applying Existing Classifiers: Two existing classifiers, Naive Bayes and Support Vector Machine (SVM), are implemented for intrusion detection. Each classifier is trained on the preprocessed data, and their performances are evaluated using accuracy, precision, recall, and F1-score metrics.

Performance Evaluation and Comparison: The code calculates and displays performance metrics such as accuracy, precision, recall, and F1-score for both the proposed ECNN model and the existing classifiers. Confusion matrices are visualized to provide insights into the classification results.

Prediction on Test Data: Finally, the code allows users to upload a new dataset for prediction using the trained ECNN model. The selected dataset undergoes the same preprocessing steps, and the model predicts the attack types for each record. The results are displayed in the Tkinter GUI

3.1 EECNN

According to the facts, training and testing of ECNN involves in allowing every source data via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from. Convolution layer is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here d=3 since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(k_x, k_y, d)$..

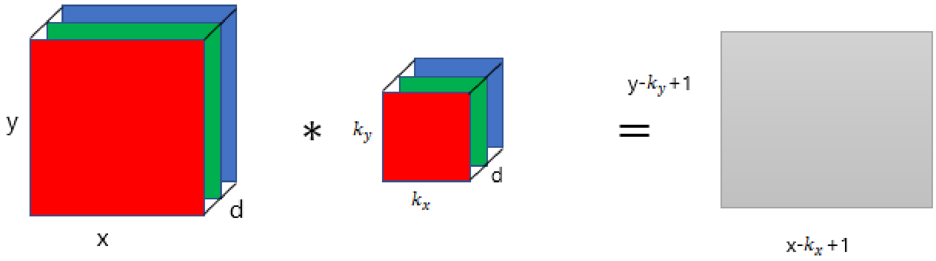
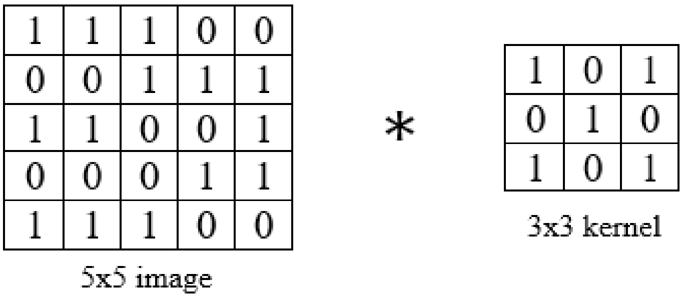


Figure 2: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\left((x - k_x + 1), (y - k_y + 1), 1\right)$, which is referred as feature map. Let us assume an input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values.



1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

(a)

$$\begin{bmatrix}1&1&1&0&0\\0&0&1&1&1\\1&1&0&0&1\\0&0&0&1&1\\1&1&1&0&0\end{bmatrix} * \begin{bmatrix}1&0&1\\0&1&0\\1&0&1\end{bmatrix} = \begin{bmatrix}3&3&3\\2&2&3\\3&2&3\end{bmatrix}$$

5x5 image 3x3 kernel Feature map

(b)

Figure 3: Example of convolution layer process (a) an image with size 5×5 is convolving with 3×3 kernel (b) Convolved feature map.

ReLU layer: Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

Max pooing layer: This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

4. Results and discussion

Figure 3 presents confusion matrices for the results obtained using different classifiers: Naïve Bayes, SVM, and the proposed ASS with ECNN model. Confusion matrices provide detailed information about the model's performance, showing the true positive, true negative, false positive, and false negative predictions for each class. Table 1 presents a comparison of various performance metrics (such as accuracy, precision, recall, and F1-score) obtained from different models. These metrics provide insights into the effectiveness of each model in detecting network intrusions.

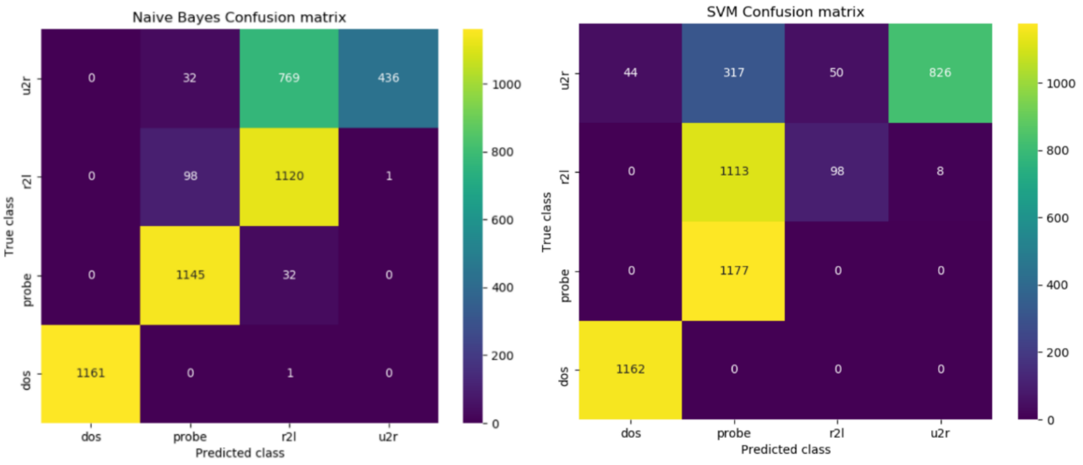


Figure 3: Obtained confusion matrices of Naïve Bayes and SVM classifier models.

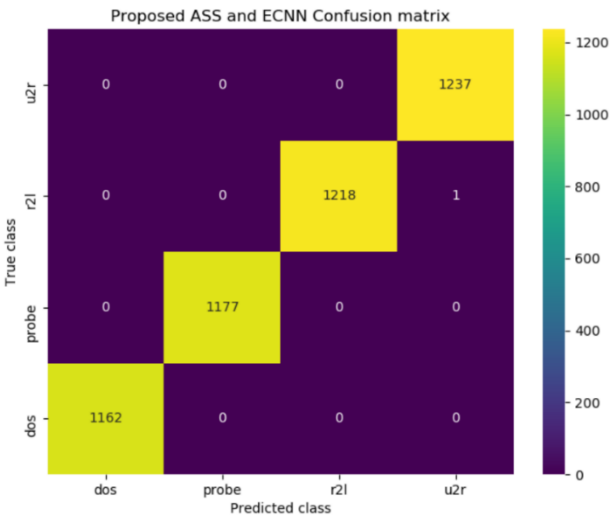
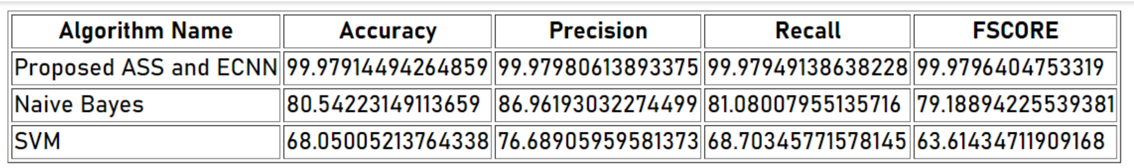


Figure 4: Confusion matrix obtained using existing naïve bayes, SVM classifiers and proposed ASS with ECNN model.

Figure 4 and Table 1 displays a graphical representation of the performance evaluation results for existing models (such as Naïve Bayes and SVM) compared with the proposed ASS and ECNN model. The graph could show trends or comparisons of metrics across different models, providing a visual summary of the evaluation results.

Table 1: Performance comparison of obtained quality metrics for wireless network intrusion detection system.



Algorithm Name	Accuracy	Precision	Recall	FSCORE
Proposed ASS and ECNN	99.97914494264859	99.97980613893375	99.97949138638228	99.9796404753319
Naive Bayes	80.54223149113659	86.96193032274499	81.08007955135716	79.18894225539381
SVM	68.05005213764338	76.68905959581373	68.70345771578145	63.61434711909168

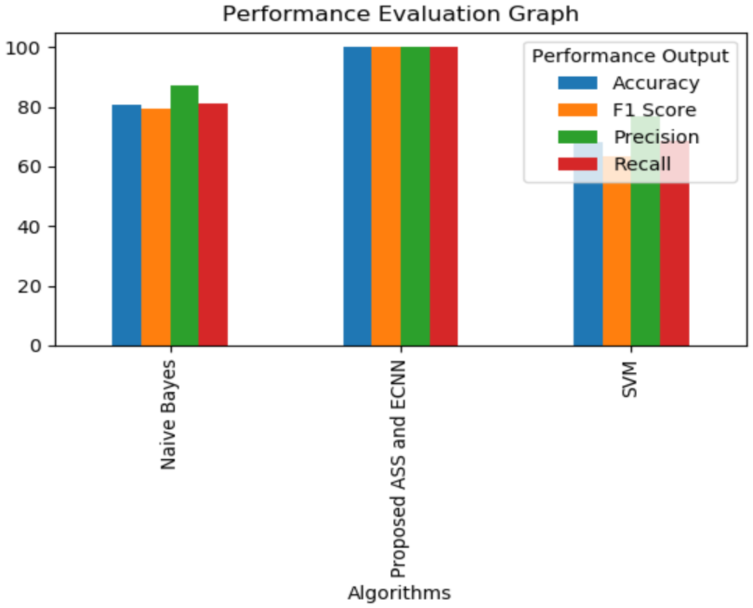


Figure 5: Performance evaluation graph of existing and proposed models.

5. Conclusion

This project successfully implemented a wireless network IDS by integrating the ASS technique with the ECNN model. This hybrid approach proved to be highly effective in addressing the class imbalance challenge commonly encountered in intrusion detection datasets. The system demonstrated superior performance compared to traditional classifiers like Naive Bayes and SVM. The ASS with ECNN model outperformed its counterparts, showcasing higher accuracy, which is vital for precise identification of network anomalies. Moreover, the model exhibited superior precision, indicating a reduced rate of false positives, and higher recall, indicating fewer false negatives. The F1-score, representing a balanced trade-off between precision and recall, highlighted the system's efficiency in handling both types of classification errors. These findings emphasize the model's effectiveness in accurate intrusion detection. In addition, these enhanced metrics underscore the system’s robustness in accurately distinguishing between normal and intrusive network activities. In conclusion, the wireless network IDS is poised to evolve, adapting to emerging threats, and maintaining its effectiveness in safeguarding network infrastructures. The ongoing research and implementation efforts will contribute significantly to the field of network security, ensuring robust and proactive intrusion detection capabilities. Looking ahead, there are several avenues for further improvement and exploration. One direction involves real-time intrusion detection, allowing the system to respond promptly to potential threats as they occur. Additionally, incorporating advanced feature engineering techniques could enhance the model's ability to capture intricate patterns in network data. Exploring ensemble learning methods, such as Random Forest or Gradient Boosting, presents an opportunity to harness the combined strengths of multiple algorithms.

References

[1] P.-F. Wu and H.-J. Shen, “The research and amelioration of patternmatching algorithm in intrusion detection system,” in Proc. IEEE 14th Int. Conf. High Perform. Comput. Commun. IEEE 9th Int. Conf. Embedded Softw. Syst., Liverpool, U.K., Jun. 2012, pp. 1712–1715, doi: 10. 1109/HPCC.2012.256.

[2] V. Dagar, V. Prakash, and T. Bhatia, “Analysis of pattern matching algorithms in network intrusion detection systems,” in Proc. Int. Conf. Adv. Comput., 2016, pp. 1–5.

[3] I. S. Thaseen and C. A. Kumar, “Intrusion detection model using fusion of chi-square feature selection and multi class SVM,” J. King Saud Univ.- Comput. Inf. Sci., vol. 29, no. 4, pp. 462–472, Oct. 2017.

[4] B. Ingre, A. Yadav, and A. K. Soni, “Decision tree based intrusion detection system for NSL-KDD dataset,” in Proc. Int. Conf. Inf. Commun. Technol. Intell. Syst., Ahmedabad, India, 2017, pp. 207–218, doi: 10.1007/978- 3-319-63645-0_23.

[5] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. S. Kumar, M. Selvi, and K. Arputharaj, “Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks,” IET Commun., vol. 14, no. 5, pp. 888–895, Mar. 2020, doi: 10. 1049/iet-com.2019.0172.

[6] M. A. Jabbar, R. Aluvalu, and S. S. Satyanarayana Reddy, “Intrusion detection system using Bayesian network and feature subset selection,” in Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC), Coimbatore, India, Dec. 2017, pp. 1–5, doi: 10.1109/ICCIC.2017.8524381.

[7] T.-T.-H. Le, J. Kim, and H. Kim, “An effective intrusion detection classifier using long short-term memory with gradient descent optimization,” in Proc. Int. Conf. Platform Technol. Service (PlatCon), Busan, South Korea, Feb. 2017, pp. 1–6, doi: 10.1109/PlatCon.2017.7883684.

[8] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, “BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset,” IEEE Access, vol. 8, pp. 29575–29585, 2020, doi: 10.1109/ACCESS.2020. 2972627.

[9] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, “An optimization method for intrusion detection classification model based on deep belief network,” IEEE Access, vol. 7, pp. 87593–87605, 2019, doi: 10. 1109/ACCESS.2019.2925828.

[10] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning, and J. Xu, “Malicious network traffic detection based on deep neural networks and association analysis,” Sensors, vol. 20, no. 5, p. 1452, Mar. 2020.

[11] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Applying convolutional neural network for network intrusion detection,” in Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI), Udupi, India, Sep. 2017, pp. 1222–1228, doi: 10.1109/ICACCI.2017.8126009.

[12] Y. Ding and Y. Zhai, “Intrusion detection system for NSL-KDD dataset using convolutional neural networks,” in Proc. 2nd Int. Conf. Comput. Sci. Artif. Intell. (CSAI), 2018, pp. 81–85.

[13] H. Yang and F. Wang, “Wireless network intrusion detection based on improved convolutional neural network,” IEEE Access, vol. 7, pp. 64366–64374, 2019, doi: 10.1109/ACCESS.2019.2917299.

[14] Q. Zhang, Z. Jiang, Q. Lu, J. Han, Z. Zeng, S.-H. Gao, and A. Men, “Split to be slim: An overlooked redundancy in vanilla convolution,” 2020, arXiv:2006.12085. [Online]. Available: http://arxiv.org/abs/ 2006.12085

[15] L. Dhanabal and S. P. Shantharajah, “A study on NSL-KDD dataset for intrusion detection system based on classification algorithms,” Int. J. Adv. Res. Comput. Commun. Eng., vol. 4, no. 6, pp. 446–452, 2015