eISSN: 2976-1190

Journal of Engineering and Sciences

Vol. 4, Issue 2, Nov 2025, 154-170

DOI:https://doi.org/10.3126/jes2.v4i2.82057

Journal homepage: Journal of Engineering and Sciences (nepjol.info)

Publisher: Research Management Cell, Pashchimanchal Campus, Lamachour, Pokhara

Intrusion Detection System using Clustering, Deep Learning, and Feature Reduction Technique

Ashish KC Khatri[1,*], Bidur Devkota[1], Sujan Tamrakar[1]

[1]*Gandaki College of Engineering and Science, Pokhara University*

[*]Corresponding author: ashishkc404@gmail.com

(Manuscript Received: 20/07/2025; Revised: 28/11/2025; Accepted: 30/11/2025)

Abstract

This research proposes an intrusion detection system (IDS) leveraging a hybrid machine learning and deep learning approach to enhance multiclass classification accuracy. The study uses the KDD99 dataset, comprising 494,020 instances with 42 features, which undergoes preprocessing steps including label encoding, memory optimization (reducing the data size by 49.99%), and feature scaling using a Standard Scaler. A hybrid feature selection technique combining Select Best and Recursive Feature Elimination (RFE) reduces the feature set from 42 to 15, improving computational efficiency. For clustering, K-means (with optimal *k* determined via the elbow method) and DBSCAN (with epsilon tuning using the knee method) are employed. Performance is evaluated using the Silhouette Score, where K-means achieves the highest score (0.885) with 11 clusters. The clustered data is then classified using an LSTM model, achieving 99.92% accuracy for 5 clusters, 99.86% for 11 clusters, and 99.53% for 22 clusters. The model demonstrates high precision, recall, and F1-scores across all clusters, even for minority attack classes. Comparative analysis with existing methods highlights the superiority of the proposed hybrid feature selection and LSTM-based classification approach. Future work may explore alternative clustering techniques and real-time deployment for enhanced intrusion detection performance.

Keywords: Intrusion Detection System (IDS), K-means, DBSCAN, LSTM, Feature Selection, Silhouette Score, KDD99

1. Introduction

1.1 Background

In an era characterized by the unprecedented spread of digital devices, the seamless integration of smart technologies, devices, and networks into our daily lives brings about unparalleled convenience. However, this interconnected landscape also exposes us to a rising threat landscape, as cyber adversaries exploit vulnerabilities in digital ecosystems. The imperative to secure the devices and networks against malicious attacks has become more pressing than ever. This research aims to address the growing security concerns surrounding IoT devices by leveraging machine learning (ML) techniques. Traditional security measures often fall short in adapting to the dynamic nature of IoT environments, necessitating the exploration of innovative approaches.

Intrusion detection systems (IDS) have emerged as a critical line of defense against the evolving spectrum of cyber threats targeting modern networks, including the Internet of Things (IoT), cloud environments, and smart systems. With the proliferation of high-volume, high-velocity data, traditional

signature-based detection methods fall short in identifying novel or zero-day attacks. This has driven the adoption of intelligent, data-driven approaches such as machine learning (ML) and deep learning (DL), enabling adaptive, scalable, and robust intrusion detection mechanisms.

A key foundation for this study proposed using a Hybrid Unsupervised Clustering-Based Anomaly Detection Method that combines K-means and DBSCAN clustering techniques to enhance anomaly detection accuracy in complex network environments (Pu et al., n.d.). Their model efficiently merges unsupervised learning with clustering refinement, addressing issues of poor cluster separation and sensitivity to parameter tuning in traditional clustering approaches. By integrating unsupervised anomaly detection with ensemble strategies, their framework demonstrates improved detection of outliers and novel attacks, making it a strong baseline for future IDS enhancements.

1.2 Literature Survey

Several studies have followed this hybrid philosophy by combining clustering with advanced deep learning or ensemble techniques. The research explored the efficacy of BiLSTM for detecting Distributed Denial of Service (DDoS) attacks in IoT, concluding that sequential DL models outperform conventional classifiers, especially in multiclass classification scenarios (Esmaeili et al., 2022). Similarly, the applied CNN, DNN, and RNN models for detecting DDoS in Agriculture 4.0 environments, with CNN and RNN excelling in different contexts depending on attack structure and classification type (Ferrag et al., 2021).

Addressing the issue of data imbalance and feature redundancy, a hybrid CNN-GRU model was introduced that integrates adaptive sampling (ADASYN and RENN), feature selection using Random Forest and Pearson correlation, and an attention mechanism for improving performance on NSL-KDD and CIC-IDS2017 datasets (Cao et al., 2022). The use of attention and multi-pooling strategies helped reduce computational complexity and enhance accuracy in multiclass intrusion classification.

The fusion of CNNs and LSTMs has also gained traction, with a CNN-LSTM-based IDS using batch normalization and dropout layers. Their model showed high accuracy and a low false alarm rate (FAR) across CICIDS2017, UNSW-NB15, and WSN-DS datasets (Halbouni et al., 2022). This combination exploits CNNs' spatial feature learning and LSTMs' temporal sequence modeling, creating a robust framework for detecting sequential attack patterns.

An alternative yet effective route has been explored through ensemble and dynamic classification models. A dynamic multi-class classifier was proposed that uses a selection system to match the most suitable ML model with specific attack types, enhancing detection accuracy over static models (Larriva-Novo et al., 2020). Likewise, integrated feature selection via XGBoost with DNNs showed superior performance over traditional shallow ML models using NSL-KDD (Devan & Khare, 2020).

Clustering remains a fundamental unsupervised learning approach for IDS. While K-means is widely used for its simplicity, its limitations in handling nonlinear boundaries led researchers to explore kernel-based and density-based alternatives. An automatic parameter-tuning method for DBSCAN was proposed using k-distance graphs, significantly improving performance across datasets with varying density (Starczewski et al., 2020). Meanwhile, utilizing K-means clustering on the NSL-KDD dataset emphasized the need to optimize the number of clusters for better classification efficiency and reduced false positives (Duque & Omar, 2015).

Another significant dimension in IDS research is feature selection. A hybrid method combining Mutual Information (MI) and Recursive Feature Elimination (RFE), validated across UCI datasets using a Random Forest classifier. Their method outperformed individual feature selection techniques in terms of F1-score, precision, recall, and accuracy (Venkatesh & Anuradha, 2019). Similarly, research focused

on feature reduction using RFE and ChiSqSelector, respectively, to enhance ML models such as SVM and RF in Big Data environments like Apache Spark (Patgiri et al., 2018) (Othman et al., 2018).

Clustering evaluation has also been refined through metrics such as the Silhouette index. By replacing a supervised evaluation with an unsupervised Silhouette-based weighting function to enhance kernel K-means clustering, aiding IDS in the absence of labeled data (Shutaywi & Kachouie, 2021). Further examining the distance metrics in the elbow and silhouette methods, noting limited impact but highlighting the Manhattan distance variability (Saputra et al., 2020).

A notable recent advancement is the emphasis on realistic, up-to-date datasets. The CIC-DDoS2019 dataset was developed to address limitations in existing IDS datasets. Their work also introduced a feature-weighting method to optimize DDoS attack classification, which serves as a valuable resource for benchmarking IDS models (Sharafaldin et al., 2019).

The research focused on LSTM with PCA and Mutual Information for dimensionality reduction was tested on KDD99; the approach proved robust in both binary and multiclass classifications, with PCA delivering the best performance (Laghrissi et al., 2021). A systematic review of machine- and deep learning-based IDSs, identifying key challenges including class imbalance, detection latency, and dataset suitability, was contributed. They offered a taxonomy of ML/DL methods and proposed directions for addressing limitations (Ahmad et al., 2021).

On the hardware front, GPU-based acceleration of anomaly-based IDS was explored, achieving up to 185x speedup in attribute calculation and 13x in session resolution over CPU-based approaches (Shrestha et al., 2019). Their findings support hardware-aware deployment of IDS in high-throughput environments.

Furthermore, to enhance the detection rate, the proposed research will combine the techniques of two different feature selection methods, filter-based and wrapper-based. This hybrid combination will strengthen feature selection, enhancing intrusion detection on the most popular dataset, KDD99. This research evaluates the performance of the hybrid feature selection methods and the K-means algorithm for a multi-labeled dataset. The study will utilize diverse datasets containing normal network traffic and attack instances on networking devices for training and evaluating a deep learning-based intrusion detection system.

1.3 Contribution

The proposed work empirically illustrates the use of clustering and LSTM models for a multi-class intrusion detection system. Listed below are the contributions accomplished by this study.

- Proposed a novel combination of filter-based (SelectKBest) and wrapper-based (Recursive Feature Elimination) feature selection techniques to enhance intrusion detection accuracy while reducing computational overhead.
- Applied K-means clustering with an elbow method for optimal cluster determination, enabling better categorization of attack types (both known and unknown) before classification.

2. Materials and Method

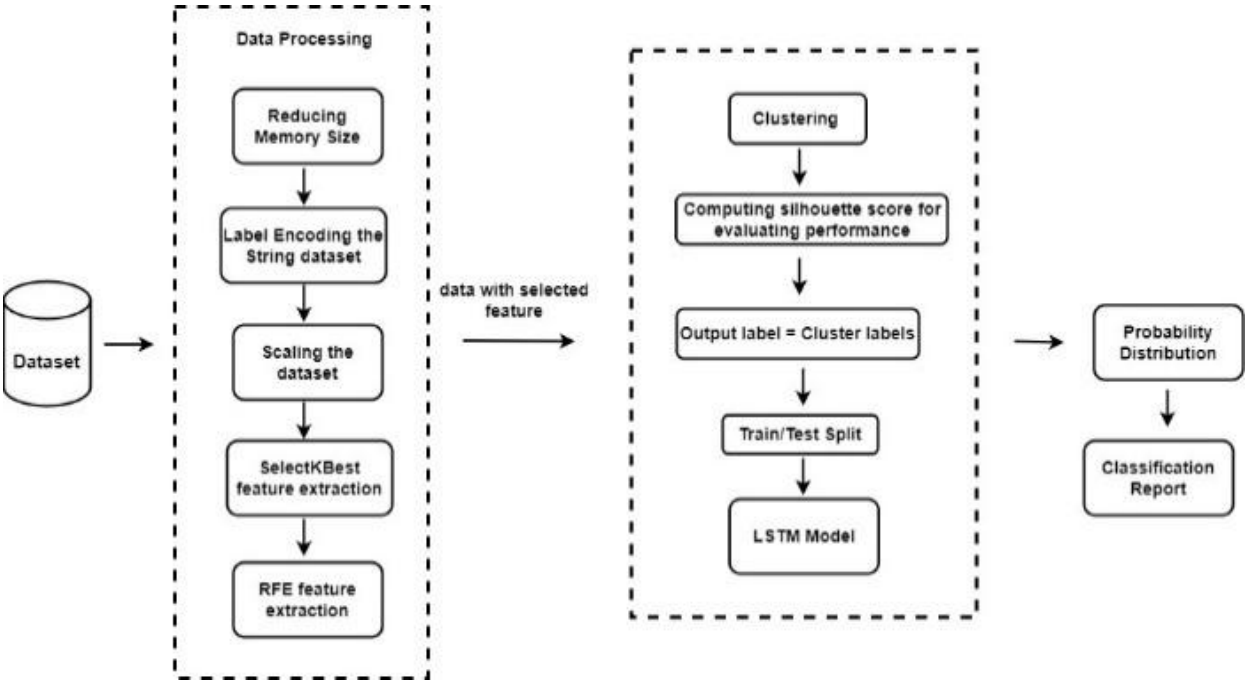

Figure 1 Overall Methodology

Figure 1: Illustrate the overall methodology for the intrusion detection system.

2.1 Dataset

The proposed model will use the open-source UCI Machine Learning Archive dataset, i.e., the KDD99 Cup dataset, which contains 42 features and 494,020 rows.

2.2 Techniques and Algorithms

2.2.1 Label Encoding and Memory Reduction

There are 42 feature attributes with int64, object and float64. The object features were then converted to numerical values using the label encoder function. This function allocates each string value a unique sequence of numbers. Once all the data were numerical, the overall dataset size was reduced by 49.99% to avoid high memory usage during model computation.

2.2.2 Scaling

The proposed research will classify output labels into multiple classes; therefore, the dataset needs to be scaled down so that all attributes fall within a similar numeric range. For this purpose, the Standard Scalar function will be used. Standard scaling uses mean and standard deviation to compute the standard score (also called the z score).

2.2.3 Feature Selection

While traditional feature extraction techniques, such as Principal Component Analysis (PCA), effectively reduce dimensionality by creating new, uncorrelated components, they operate in an unsupervised manner and may not align with the specific objective of maximizing classification performance. Similarly, conventional filter methods, though computationally efficient, select features based on generalized statistical metrics without regard for the classifier's bias, often resulting in suboptimal accuracy.

In contrast, wrapper methods explicitly optimize classification performance by evaluating feature subsets against the target classifier, but at prohibitive computational cost and the risk of overfitting, especially with high-dimensional data. To bridge this gap, a hybrid filter-wrapper framework is proposed. This approach leverages the filter method's speed for an initial, aggressive feature reduction, creating a tractable subset of promising candidates. A wrapper method is then applied to this refined subset to perform a targeted optimization. This synergistic strategy mitigates the primary weakness of filter methods (low classifier-specific accuracy) and the computational intractability of wrapper methods, while outperforming PCA by ensuring the final feature set is directly relevant to the classification task, rather than merely representing the axes of most significant variance (Venkatesh & Anuradha, 2019).

For feature selection, the research model used the combination of SelectKBest and the Recursive Feature Elimination algorithm.

Table 1: Feature Selection

Description	No. of features
Original Dataset	42
After Applying SelectKBest	30
After Applying RFE	15

From the *Table 1* the final number of features that were fed into the clustering was obtained as 15.

2.2.4 Clustering

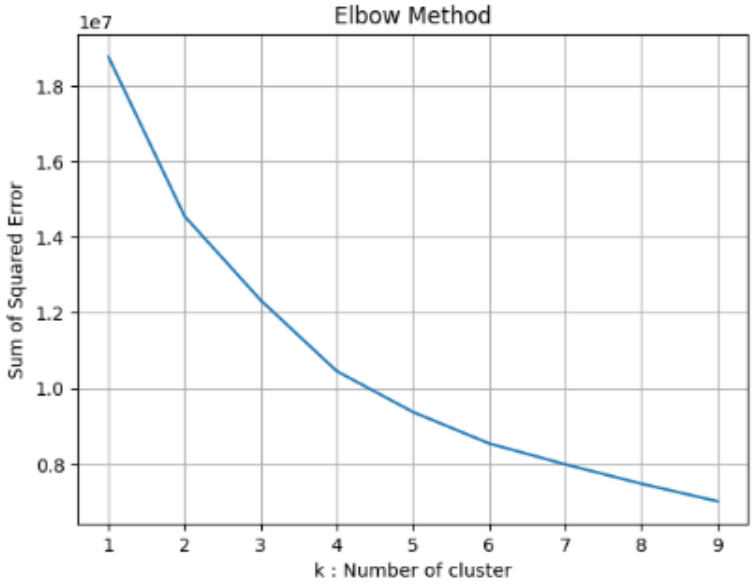

Figure 2: Elbow Method to find the number of clusters

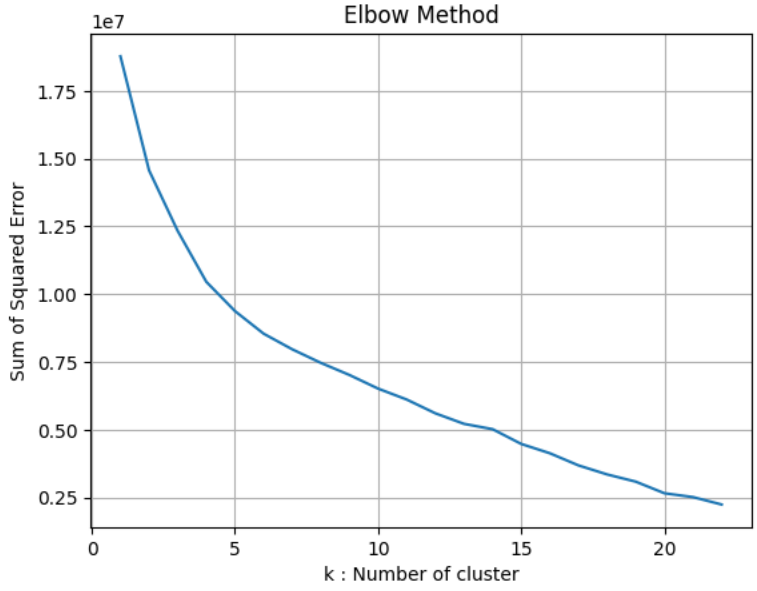

Figure 3: Overall Number of clusters

The extracted feature was then used to cluster and determine the output labels. For comparison, two of the clustering techniques were used: K-means and DBSCAN. One of the challenges with k-means is the determination of the value of k. To overcome this, the elbow method was used.

From the Figure 2 and Figure 3, the value of k can be 2, for binary classification based on sudden change in slope, or 5 for multiclass classification. The clustering algorithm was applied. The original count of attacks in the dataset and the clustered result is listed as follows in Table 2.

Table 2: Data Count

SN	Attack Type	Data Count
1	smurf	280790
2	neptune	107201
3	normal	97277
4	back	2203
5	satan	1589
6	ipsweep	1247
7	portsweep	1040
8	warezclient	1020
9	teardrop	979
10	pod	264
11	Nmap	231
12	guess_passwd	53
13	buffer_overflow	30
14	land	21
15	warezmaster	20
16	imap	12
17	rootkit	10
18	loadmodule	9
19	ftp_write	8

20	multihop	7
21	phf	4
22	perl	3

Based on the Figure 2 and Figure 3, a significant change of slope was observed on value of k for 2, 4, and 5. The value of k=4, 5 and 11 can be used for multiclass classification. The dataset already had 22 different output variables so value of k= 22 was set for one part of the research. Likewise, a summarized preview for k =11 was tested and finally, to evaluate the proposed model with all the existing research on the dataset with a value of k =5 was chosen (Talukder et al., 2023). The research has used multiple clusters using k values 5,11,22, and evaluated the performances. Likewise, the proposed DBSCAN algorithm was implemented. The DBSCAN algorithm uses two parameters: epsilon and minimum point. The epsilon value represents the radius of the cluster, and the minimum point represents the minimum value in the cluster. To compute the eps value, the knee method of k-distance method was used.

2.2.5 Silhouette Score

The silhouette score is computed as:

$$S(x_i) = \frac{b(xi) - a(xi)}{\max\{b(xi), a(xi)\}}$$

where xi is an element in cluster πk,

a(xi) is the average distance of xi to all other elements in the cluster πk (within dissimilarity),

b(xi) = min {dl(xi)}, among all clusters $l \neq k$.

where $d_l(xi)$ is the average distance from xi to all points in cluster π_l for $l \neq k$ (between dissimilarity)

The performance of the clustering algorithm will be measured using the Silhouette score. This score measures the distance between the cluster points and yields results ranging from -1 to 1. The score closest to 1 represents a good performance of the clustering algorithm. Table 3 shows the comparison of the Silhouette score of k-means with multiple clusters.

Table 3: Comparison of Clusters and Silhouette Score

SN	Number of clusters	Silhouette Score
1	5	0.831
2	11	0.885
3	22	0.852

From the Figure 4, the value of epsilon was assumed to be in the range 0 to 0.2. The experiment was performed to identify the optimal value.

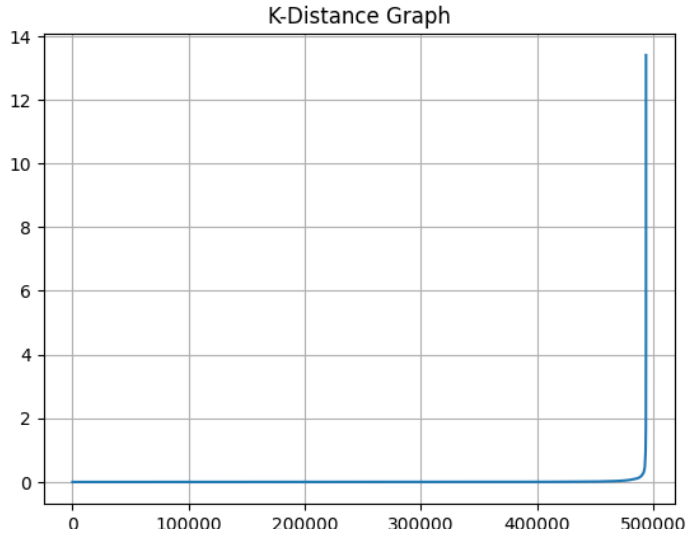

Figure 4: K-Distance Graph

The Table 4 shows the experiment performed on the DBSCAN to identify parameters for the algorithm to be used. The optimal values for these parameters are more complex. i.e, the eps radius and the MinPts density threshold. The choice of these parameters is especially difficult when the density variation within clusters is significant(Starczewski et al., 2020). Thus, multiple experiments were performed on the basis of Figure 4, but all the parameters could not give the optimal result for clustering, as seen in the Silhouette score. This is because of the diverse data distribution for cluster values.

Table 4: Experiment on DBSCAN

SN	Method	Eps value	Minpoint	Silhouette Score
1	2*D	0.05	30	0.32
2	2*D	0.1	30	0.344
3	D+1	0.2	16	0.392
4	D+1	0.05	16	0.519
5	Briant	0.2	13	0.404
6	Briant	0.1	13	0.411
7	Briant	0.02	12	0.461
8	Briant	0.04	13	0.482
9	Briant	0.02	13	0.518
10	Briant	0.05	14	0.548
11	Briant	0.05	13	0.566

12	Briant	0.05	12	0.569
13	Briant	0.03	13	0.519

2.2.6 LSTM algorithm

The LSTM Model consists of 3 gates: input gate, forget gate and output gate.

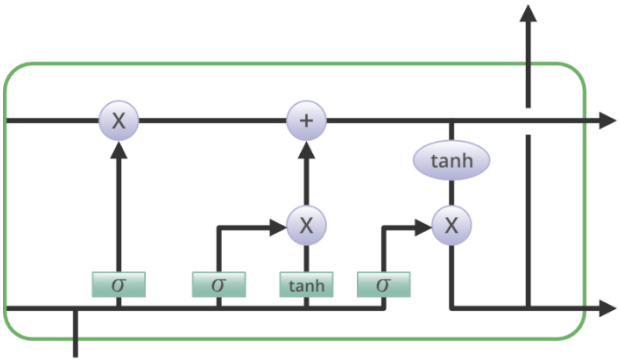

Figure 5: LSTM Model

The LSTM model used in the research is discussed below:

LSTM Layer 1: The input layer uses the same number of units as neurons in the model. This layer uses the linear shape of the input; hence, the input values are reshaped accordingly.

LSTM Layer 2: The second layer adds a dense layer to the model. The number of neurons is the same as the number of output classes. This layer uses a SoftMax activation function to produce a probability distribution for each data record across the output classes.

Model Compilation: The categorical cross-entropy function was used as the loss function, since the output labels were one-hot encoded.

Table 5 LSTM Model Architecture

Layer	Type	Units	Dropout	Recurrent Dropout	Activation	Output Shape
1	LSTM	15	0.2	0.2	Tanh + Sigmoid	(None, 15)
2	Dense	5	None	None	Softmax	(None, 5)

Table 5 Shows the overall architecture of the LSTM model used in the research.

2.2.7 Training and Validation

The dataset was first split into training and test sets at an 80:20 ratio, using stratified sampling to preserve the cluster-label distribution. For all neural network models (LSTM and DNN), the input was reshaped into a 3-dimensional tensor format to match the sequential architecture requirements.

During training, 20% of the training set was internally allocated as a validation subset using Keras' built-in validation_split parameter or via validation_data (x_test, y_test), depending on the experimental setting. This allowed continuous monitoring of validation performance during each epoch.

Each model was trained for a maximum of 100 epochs; however, to avoid unnecessary computation and overfitting, the Early Stopping callback was applied with
– monitor = 'val_loss'
– patience = 3

– restore_best_weights = True.
This ensured that training automatically stopped when the model stopped improving on the validation set, and the weights with the best validation performance were restored.

Across experiments, most models converged within 5–12 epochs, indicating effective early stopping.

Several measures were applied to control overfitting:

1. Early Stopping (primary mechanism).
2. Dropout layers (0.2 on all LSTM models with larger hidden states).
3. Normalization and feature scaling, which reduce variance and improve generalization.
4. Stratified train–test split, ensuring minority clusters were represented in both sets.

Training and validation curves were inspected for divergence; in all cases, validation loss closely tracked training loss, confirming effective overfitting control.

2.2.8 Handling Data Distribution

Due to the nature of KDD Cup99, the distribution of attack categories is highly imbalanced. This imbalance propagates into the clustering stage. Small clusters can lead to biased learning and unstable gradients during training. Therefore:

1. Stratified train–test split ensured that even very small clusters remained represented in both training and testing sets.

2. Clusters with fewer than three samples were discarded, as they did not provide enough data for meaningful learning.

3. Minority clusters were preserved during one-hot encoding, preventing them from being implicitly merged or dropped.

4. The use of LSTM, which learns sequential representations, naturally reduces noise by focusing on temporal/feature correlations rather than frequency alone.

In intrusion detection, small clusters often correspond to rare attacks that are semantically important despite low frequency. Therefore, instead of oversampling, we chose to preserve cluster purity while using early stopping and dropout to avoid overfitting to these rare samples. Hence, the number of clusters for 5, 11, and 22 was tested for the research to compare the performance.

3. Results and Discussion

The LSTM model was trained and tested using the feature sets derived from multiple K-means clustering configurations (k=5, 11, 22). Early stopping was implemented to halt training if validation loss plateaued for three consecutive epochs, ensuring computational efficiency without sacrificing performance.

3.1 LSTM results 5 clusters

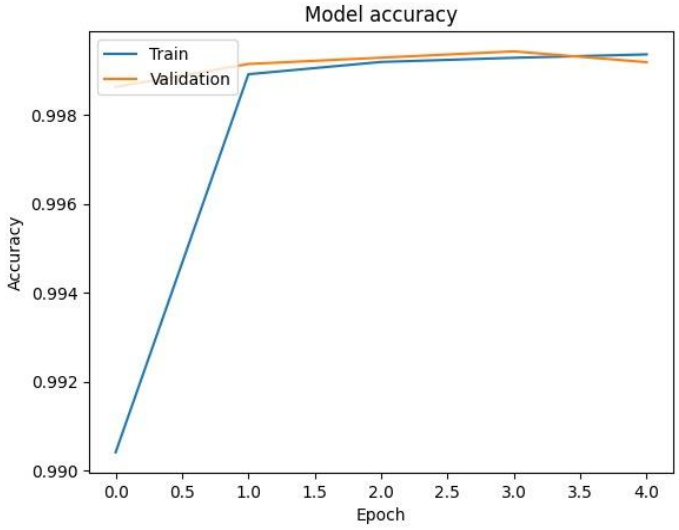

Figure 6: Model Accuracy for 5 clusters

The model achieved an overall accuracy of 99.92% in just five epochs, with a final loss of 0.0021. As shown in Table 6, the classification report demonstrates near-perfect precision, recall, and F1-scores (1.00) across all clusters, except for Cluster 4 (F1-score: 0.99), indicating robust generalization.

The classification report of the performance of the LSTM model under K-means with k=5 is:

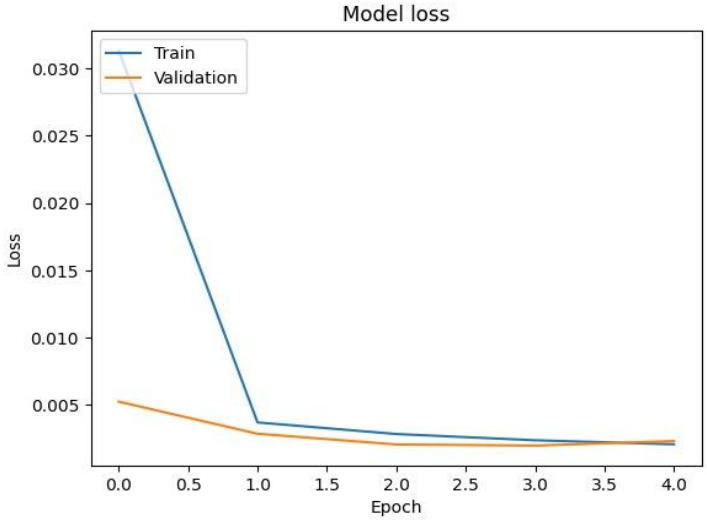

Figure 7: Model loss for 5 clusters

Table 6: Performance metrics for 5 clusters

Cluster Label	Precision	Recall	F1 Score	Support
0	1.00	1.00	1.00	56145
1	1.00	1.00	1.00	5198
2	1.00	1.00	1.00	17352
3	1.00	1.00	1.00	18194

4	1.00	0.98	0.99	1915

3.2 LSTM results with 11 clusters

Accuracy slightly decreased to 99.86% with 11 clusters, though performance remained strong. *Table 7*

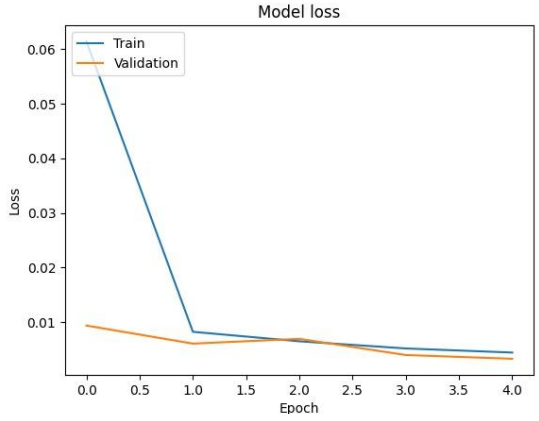

Figure 8: Model loss for 11 clusters

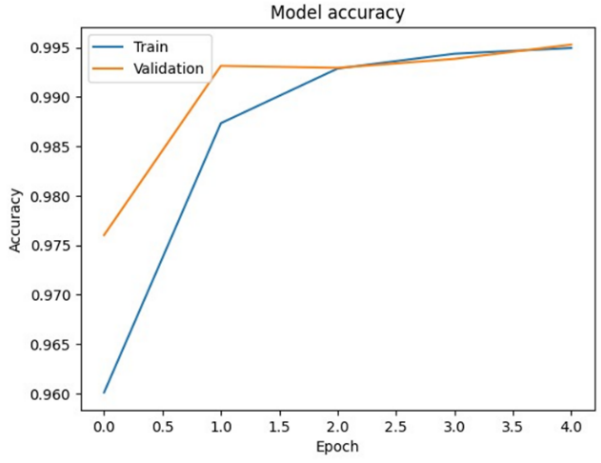

Figure 9: Model accuracy for 11 clusters

reveals high scores (≥0.99) for most clusters, with minor declines in smaller clusters (e.g., Cluster 4: F1-score 0.83; Cluster 6: 0.94), likely due to limited support (n=5 and n=86, respectively).

The classification report of the performance of LSTM model under K-means with k=11 is:

Table 7: Performance metrics for 11 clusters

Cluster Label	Precision	Recall	F1 Score	Support
0	1.00	1.00	1.00	56144
1	1.00	1.00	1.00	17331
2	1.00	1.00	1.00	17548
3	0.98	1.00	0.99	1225
4	0.71	1.00	0.83	5
5	1.00	1.00	1.00	4247
6	0.93	0.95	0.94	86
7	0.99	1.00	0.99	1129
8	0.99	0.99	0.99	384

9	1.00	1.00	1.00	292
10	0.99	0.98	0.98	413

3.3 LSTM results with 22 clusters

At k=22, accuracy further declined to 99.53%, reflecting increased complexity. Table 8 shows excellent performance (F1-score ≥0.95) for larger clusters (e.g., Clusters 0–2), while smaller clusters (e.g., Cluster 5: 0.60, Cluster 16: 0.43) exhibited variability due to sparse data.

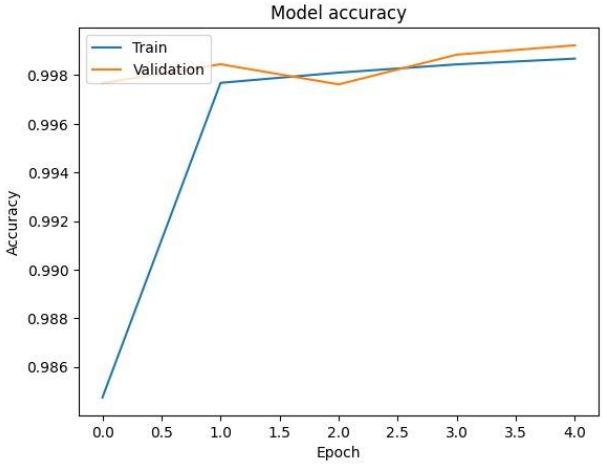

Figure 10: Model accuracy for 22 clusters

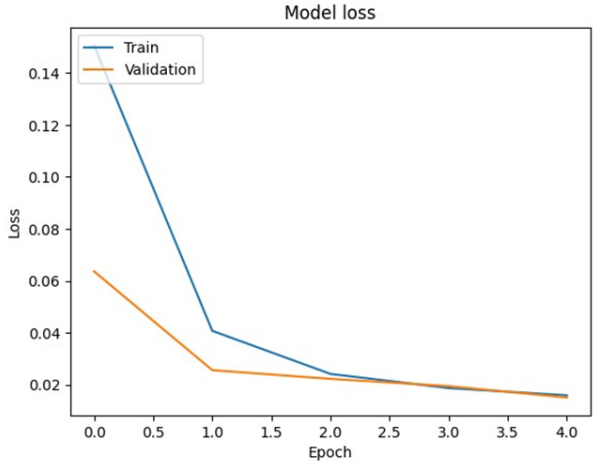

Figure 11: Model loss for 22 clusters

The classification report of the performance of the LSTM model for 22 clusters is shown in Table 8.

Table 8: Performance metrics for 22 clusters

Cluster Label	Precision	Recall	F1 Score	Support
0	0.99	1.00	1.00	7679
1	1.00	1.00	1.00	56137
2	1.00	1.00	1.00	17328
3	0.98	0.96	0.97	536
4	0.99	0.97	0.98	724
5	0.50	0.75	0.60	4
6	0.78	0.98	0.87	63
7	1.00	1.00	1.00	4173
8	0.96	0.96	0.96	222
9	0.99	0.99	0.99	307
10	0.94	0.97	0.95	260
11	1.00	0.33	0.50	6
12	1.00	0.93	0.96	71
13	0.99	0.99	0.99	3949
14	0.98	0.96	0.97	2879
15	0.99	1.00	1.00	2446
16	0.62	0.33	0.43	24
17	0.96	0.92	0.94	252
18	0.96	1.00	0.98	495
19	0.95	0.94	0.94	271
20	0.95	0.93	0.94	485

3.4 Comparative Analysis

Table 9 benchmarks our results against prior work. The proposed hybrid approach (SelectKBest + RFE for feature selection, K-means + LSTM for modeling) outperforms existing methods, achieving 99.92% accuracy (k=5)—a 8–38% improvement over comparable studies (e.g., Halbouni et al., 2022: 82%; Duque & Omar, 2015: 81.61% for k=11).

Table 9: Comparative Performance of State-of-the-Art Methods

SN	Paper	Feature Selection	Algorithm	Accuracy (in %)
1	(Pu et al., n.d.)	F-Test	SSC-OCSVM	91, 89
2	(Halbouni et al., 2022)	SelectKBest	CNN-LSTM	82
3	(Duque & Omar, 2015)		Kmeans	81.61 (11 clusters) 65.40 (22 clusters) 61.30 (44 clusters) 55.43 (88 clusters)
4	(Larriva-Novo et al., 2020)	Correlation Ken-dall coefficient	KNN, SVM, DT, RF, XGBOOST, MLP, LSTM	77.9, 73.9, 80.1, 82.8, 82.4, 81.1, 81.6

5	Proposed Research	Hybrid (SelectKBest and RFE)	Kmeans for clustering LSTM for classification	99.92(5 clusters) 99.86(11 clusters) 99.53(22 clusters)

4. Conclusions and Future Works

4.1 Conclusions

The proposed work introduces an intrusion detection framework that integrates hybrid feature selection (SelectKBest and RFE), unsupervised clustering, and an LSTM-based deep learning classifier to model attack behaviors more effectively. The hybrid feature selection strategy enabled the model to retain only the most discriminative attributes, thereby reducing noise and improving learning efficiency.

By combining optimized clustering with sequential deep learning, the system automatically grouped diverse attack patterns and improved multi-class classification performance. The use of the elbow method addressed the standard limitation of the k-means algorithm, which requires manual selection of k, enabling a more data-driven, stable clustering process.

Experimental results demonstrate that the model generalizes well across different cluster configurations and maintains strong performance even on minority and small-sized clusters. This indicates that the framework is robust to variations in cluster density and distribution—an important requirement in real-world intrusion detection scenarios where rare attacks are highly impactful.

Overall, the research highlights three significant contributions:

1. A hybrid feature selection pipeline that improves model interpretability and reduces computational overhead.
2. A clustering-enhanced classification approach that supports automated attack aggregation and improves detection of multi-class intrusion categories.
3. A deep learning model capable of handling imbalanced and small clusters, demonstrating resilience in scenarios involving rare or emerging attack types.

These findings suggest that the proposed methodology can serve as a practical foundation for next-generation intrusion detection systems, particularly in environments where attack behaviors are evolving and inherently imbalanced.

4.2 Future Works

The combination of clustering techniques and deep learning models has resulted in good performance. However, for future work, the silhouette coefficient for another clustering technique, DBSCAN, can be improved with a more refined epsilon and minimum point values. The Briant method was applied for computation, but the optimal value was still not met. This can lead to future research and significantly improve clustering results. Moreover, future work can include applying an optimal clustering technique to a hybrid model for new types of attacks.

Acknowledgment

This work is supported by the Faculty of Science and Technology, Gandaki College of Engineering and Science, Pokhara, Nepal.

References

[1] Pu, G., Wang, L., Shen, J., & Dong, F. (n.d.). *A hybrid unsupervised clustering-based anomaly detection method*. http://creativecommons.org/licenses/by/4.0/

[2] Esmaeili, M., Goki, S. H., Masjidi, B. H. K., Sameh, M., Gharagozlou, H., & Mohammed, A. S. (2022). ML-DDoSnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and NSL-KDD. *Wireless Communications and Mobile Computing, 2022*, Article 8481452. https://doi.org/10.1155/2022/8481452

[3] Ferrag, M. A., Shu, L., Djallel, H., & Choo, K. K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics, 10*(11). https://doi.org/10.3390/electronics10111257

[4] Cao, B., Li, C., Song, Y., Qin, Y., & Chen, C. (2022). Network intrusion detection model based on CNN and GRU. *Applied Sciences, 12*(9). https://doi.org/10.3390/app12094184

[5] Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: Hybrid deep neural network for network intrusion detection system. *IEEE Access, 10*, 99837–99849. https://doi.org/10.1109/ACCESS.2022.3206425

[6] Larriva-Novo, X., Sánchez-Zas, C., Villagrá, V. A., Vega-Barbas, M., & Rivera, D. (2020). An approach for the application of a dynamic multi-class classifier for network intrusion detection systems. *Electronics, 9*(11), 1–18. https://doi.org/10.3390/electronics9111759

[7] Devan, P., & Khare, N. (2020). An efficient XGBoost–DNN-based classification model for network intrusion detection system. *Neural Computing and Applications, 32*(16), 12499–12514. https://doi.org/10.1007/s00521-020-04708-x

[8] tarczewski, A., Goetzen, P., & Er, M. J. (2020). A new method for automatic determining of the DBSCAN parameters. *Journal of Artificial Intelligence and Soft Computing Research, 10*(3), 209–221. https://doi.org/10.2478/jaiscr-2020-0014

[9] Duque, S., & Bin Omar, M. N. (2015). Using data mining algorithms for developing a model for intrusion detection system (IDS). *Procedia Computer Science, 61*, 46–51. https://doi.org/10.1016/j.procs.2015.09.145

[10] Venkatesh, B., & Anuradha, J. (2019). A hybrid feature selection approach for handling high-dimensional data. In *Lecture Notes in Networks and Systems* (Vol. 74, pp. 365–373). Springer. https://doi.org/10.1007/978-981-13-7082-3_42

[11] Patgiri, R., Varshney, U., Akutota, T., & Kunde, R. (2018). An investigation on intrusion detection system using machine learning. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1684–1691). IEEE. https://doi.org/10.1109/SSCI.2018.8628676

[12] Othman, S. M., Ba-Alwi, F. M., Alsohybe, N. T., & Al-Hashida, A. Y. (2018). Intrusion detection model using machine learning algorithm on big data environment. *Journal of Big Data, 5*(1). https://doi.org/10.1186/s40537-018-0145-4

[13] Shutaywi, M., & Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entropy, 23*(6). https://doi.org/10.3390/e23060759

[14] Saputra, D. M., Saputra, D., & Oswari, L. D. (2020). *Advances in Intelligent Systems Research*. (Publication data incomplete)

[15] Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *International Carnahan Conference on Security Technology (CCST)*. IEEE. https://doi.org/10.1109/CCST.2019.8888419

[16] Laghrissi, F. E., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data, 8*(1). https://doi.org/10.1186/s40537-021-00448-4

[17] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies, 32*(1). https://doi.org/10.1002/ett.4150

[18] Shrestha, B., Karna, R., Shrestha, P., Bhatt, B., & Verma, A. (2019). *Performant anomaly-based network intrusion detection system using GPU*. *International Journal of Advanced Engineering*. http://ictaes.org