# Clusters in Chaos:
# A Deep Unsupervised Learning Paradigm for Network Anomaly Detection

Under the guidance of
Dr. P. Kola Sujatha

Krishnaa S (2020506045)
Jawahar A S (2020506035)
Thamizharasi M (2020506102)

## PROBLEM STATEMENT

With the ever-increasing sophistication of cyberattacks, traditional security measures are struggling to keep pace, leaving networks vulnerable to disruptions, data breaches, and financial losses. Hence, an intelligent system capable of detecting anomalous network behaviour is crucial for identifying these threats before they can cause significant damage.

# OBJECTIVES

- Analyse and understand the NSL KDD dataset.
- Develop a deep unsupervised learning model.
- Optimize the clustering techniques and evaluate the model.

# LITERATURE SURVEY

| | Description | Pros and Cons |
|---|---|---|
| 1 | **ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection**<br><br>**Published in:** IEEE Transactions on Network and Service Management , IEEE<br>**Year:** 2023<br>**Authors:** Willian Tessaro Lunardi, Martin Andreoni Lopez, Jean-Pierre Giacalone<br>**About:** ARCADE, an unsupervised deep learning detection system for network anomaly detection. ARCADE uses a convolutional Autoencoder to build a profile of normal traffic and detect potential anomalies and intrusions efficiently. | **Pros:**<br>• ARCADE achieves nearly 100% F1-score for detecting malicious traffic.<br>• ARCADE's 20 times fewer parameters result in faster detection speed.<br>• Adversarial strategy is adaptable to various autoencoder architectures.<br>**Cons:**<br>• ARCADE shows lower F1-scores (68.70% and 66.61%) for HTTP DOS, DDOS attacks.<br>• Relying on two initial packets may hinder accuracy in some cases.<br>• Computational overhead may occur during training, affecting efficiency. |
| 2 | **A Deep Learning Approach to Network Intrusion Detection**<br><br>**Published in:** IEEE Transactions on Emerging Topics in Computational Intelligence<br>**Year:** 2018<br>**Authors:** Nathan Shone , Tran Nguyen Ngoc, Vu Dinh Phai , and Qi Shi<br>**About:** Utilizes TensorFlow to implement a Deep Belief Network (DBN) for efficient network anomaly detection, tackling challenges in Network Intrusion Detection Systems (NIDS). It introduces Non-Symmetric Dimensionality Auto-Encoder (NDAE) for enhanced feature learning, outperforming DBNs, and employs stacked NDAE with Random Forest for superior classification. | **Pros:**<br>• Up to 5% accuracy improvement and 98.81% training time reduction.<br>• High precision, recall, and accuracy on KDD Cup '99 and NSL-KDD datasets.<br>• Addresses concerns in human interaction and detection accuracy for modern NIDSs.<br>**Cons:**<br>• Acknowledges imperfections, indicating potential for further refinement.<br>• Untested in handling zero-day attacks and real-world backbone network traffic. |

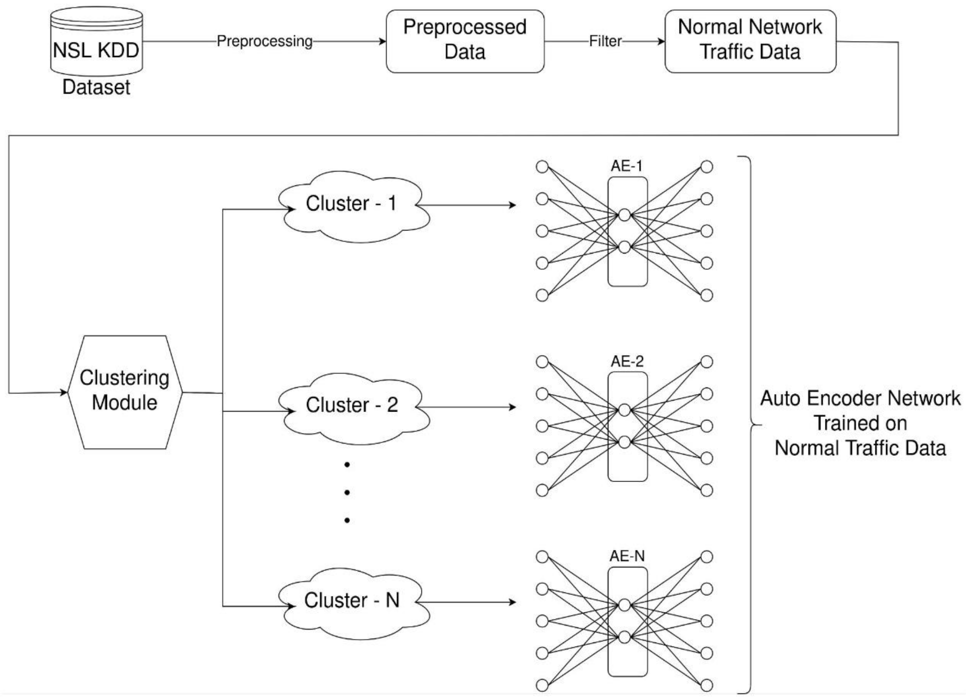
# LITERATURE SURVEY (cont.)

| | Description | Pros and Cons |
|---|---|---|
| 3 | **An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks**<br><br>**Published in:** IEEE Internet of Things Journal, Vol. 10<br>**Year:** 2023<br>**Authors:** Cheolhee Park , Jonghoon Lee , Youngsoo Kim, Jong-Geun Park , Hyunjin Kim, and Dowon Hong<br>**About:**<br>This study presents an innovative AI-based NIDS addressing data imbalance, achieving up to 93.2% accuracy on NSL-KDD. Future plans include applying the framework to federated learning and enhancing resilience against adversarial attacks in real-world environments. | **Pros:**<br>• Effectively resolves data imbalance problem in AI-based NIDS.<br>• Demonstrates significant performance improvements in detecting network threats.<br>• Future-oriented approach with plans for application in federated learning and addressing adversarial attacks.<br>**Cons:**<br>• Limited insight into potential limitations or challenges faced during implementation.<br>• Lack of specific details on the size and diversity of the real-world data set used.<br>• The study does not explicitly discuss any ethical considerations or potential drawbacks associated with the proposed AI-based NIDS.. |
| 4 | **Semisupervised-Learning-Based Security to Detect and Mitigate Intrusions in IoT Network**<br><br>**Published in:** IEEE Internet of Things Journal, Vol. 7, No. 11, IEEE<br>**Year:** 2020<br>**Authors:** Nagarathna Ravi and S. Mercy Shalinie<br>**About:** Proposing SDRK, a semi supervised learning model for IoT intrusion detection, utilizing fog nodes with 99.78% accuracy on NSL-KDD. Future work focuses on optimizing retraining time and assessing SDRK's ability to detect attacks on fog nodes. | **Pros:**<br>• SDRK achieves high 99.78% accuracy in IoT intrusion detection.<br>• Fog node placement ensures swift attack detection crucial for latency-sensitive IoT services.<br>• Semi Supervised nature addresses challenges in labeling large datasets.<br>**Cons:**<br>• Longer retraining time compared to ELM-based models may impact real-time applications.<br>• Assumed fog node immunity to attacks requires verification and additional security measures. |

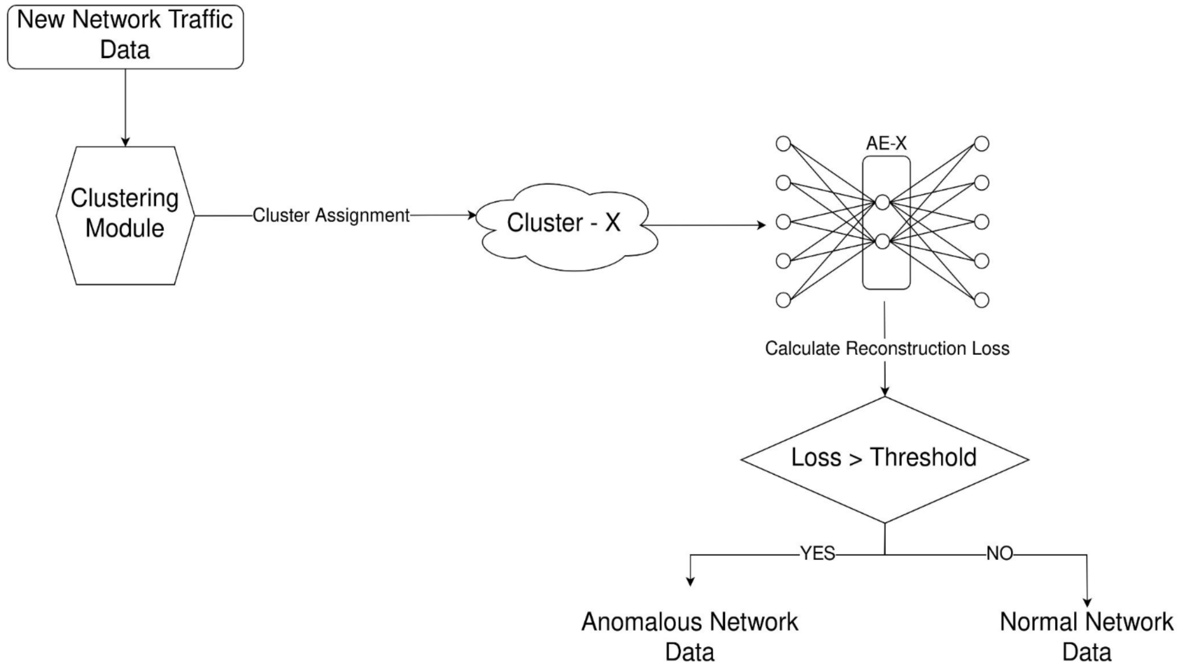
# LITERATURE SURVEY (cont.)

| | Description | Pros and Cons |
|---|---|---|
| 5 | **Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm**<br><br>**Published in:**IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 10, IEEE<br>**Year:** 2016<br>**Authors:** Mohammed A. Ambusaidi, Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan,<br>**About:** This paper introduces a mutual information-based feature selection algorithm for network intrusion detection, enhancing the performance of LSSVM-IDS. The proposed approach, FMIFS, exhibits superior results on KDD Cup 99, NSL-KDD, and Kyoto 2006+ datasets, outperforming state-of-the-art models in accuracy and computational efficiency. | **Pros:**<br>• FMIFS improves feature selection, leading to enhanced accuracy and reduced computational cost.<br>• LSSVM-IDS + FMIFS outperforms existing models in intrusion detection across diverse datasets.<br>• Achieves promising results, particularly excelling in handling U2R and R2L classes.<br>**Cons:**<br>• FMIFS algorithm optimization for search strategy could be explored in future research.<br>• Consideration of unbalanced sample distribution impact on IDS is necessary in further studies.<br>• While promising, additional research is needed to address real-world challenges and enhancements. |

# ARCHITECTURE DIAGRAM




# ARCHITECTURE DIAGRAM (cont.)



# NOVELTY

- **Clustered Autoencoders:** A novel approach using clustering to group similar normal data points, followed by training individual autoencoders for each cluster.
- **Nuanced Normal Behavior Capture:** Uniquely captures fine-grained patterns of normal behavior within each cluster, enhancing the model's ability to discern anomalies.
- **Advancements Over Standard Methods:** The approach taken represents a significant leap beyond conventional anomaly detection methods by addressing limitations, offering improved accuracy, and demonstrating versatility in handling diverse network traffic scenarios.

# ALGORITHM

**Input:** NSL KDD Dataset
**Output:** Anomaly classification for new data points

**1. Preprocessing:**
- **1.1.** Clean and preprocess the NSL KDD Dataset.
- **1.2.** Select data points labeled as normal traffic.

**2. Clustering:**
- **2.1.** Apply a clustering algorithm (e.g., K-means) to group similar normal data points.
- **2.2.** Assign each data point to the nearest cluster.

**3. Autoencoder Training:**
- **3.1.** For each cluster created in Step 2:
  - **3.1.1.** Extract data points belonging to the cluster.
  - **3.1.2.** Train a separate autoencoder on the data points of that cluster.

# ALGORITHM (cont.)

4. **Reconstruction Error Thresholding:**
   **4.1.** Calculate the reconstruction error for each new data point using the corresponding cluster's autoencoder.
   **4.2.** Set a threshold for the reconstruction error based on training data.

5. **Anomaly Detection:**
   **5.1.** Allocate new data points to the nearest pre-existing cluster using the clustering module.
   **5.2.** Predict the reconstruction error for the new data point using the corresponding cluster's autoencoder.
   **5.3.** If the reconstruction error is above the threshold, classify the data point as anomalous.
   **5.4.** If the reconstruction error is below the threshold, classify the data point as normal.

6. **Evaluation:**
   **6.1.** Assess the performance of the algorithm using appropriate metrics (e.g., precision, recall, F1 score).
   **6.2.** Fine-tune parameters based on evaluation results.

# IMPLEMENTATION

1. **Data Preprocessing**
   a. Converted attributes with categorical values into numerical values using one-hot-encoding.
   b. Used Min-Max-Scaler to properly scale attributes with widely varying numerical values.
2. **Data Analysis**
   a. Visualized the correlation matrix to identify relationships between features and understand their impact on the target variable.
   b. Extracted features with high positive and negative correlation with the target attribute.
   c. Used a Random Forest Classifier to determine the top 10 most important features in the dataset.
3. **Basic AutoEncoder Model**
   a. Built a basic AutoEncoder model with a single hidden layer.
   b. Set the dimension of the latent space to be 32 and trained the model on Normal Network Traffic.
   c. Calculated the reconstruction loss and set a threshold value.
   d. Carried out performance evaluation of the model.

# OUTPUT SCREENSHOTS

| | duration | protocol_type | service | flag | src_bytes | dst_bytes | land | wrong_fragment | urgent | hot | num_failed_logins | logged_in | num_compromis |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | tcp | ftp_data | SF | 491 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 0 | udp | other | SF | 146 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **2** | 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **3** | 0 | tcp | http | SF | 232 | 8153 | 0 | 0 | 0 | 0 | 0 | 1 | |
| **4** | 0 | tcp | http | SF | 199 | 420 | 0 | 0 | 0 | 0 | 0 | 1 | |

***NSL KDD dataset***

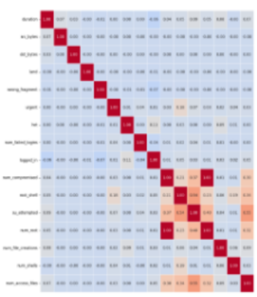
```
array(['normal', 'neptune', 'warezclient', 'ipsweep', 'portsweep',
       'teardrop', 'nmap', 'satan', 'smurf', 'pod', 'back',
       'guess_passwd', 'ftp_write', 'multihop', 'rootkit',
       'buffer_overflow', 'imap', 'warezmaster', 'phf', 'land',
       'loadmodule', 'spy', 'perl', 'saint', 'mscan', 'apache2',
       'snmpgetattack', 'processtable', 'httptunnel', 'ps', 'snmpguess',
       'mailbomb', 'named', 'sendmail', 'xterm', 'worm', 'xlock',
       'xsnoop', 'sqlattack', 'udpstorm'], dtype=object)
```

***Unique values of "label"***

# OUTPUT SCREENSHOTS (cont.)

| | duration | src_bytes | dst_bytes | land | wrong_fragment | urgent | hot | num_failed_logins | logged_in | num_compromised | root_shell |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 3.558064e-07 | 0.000000e+00 | 0 | 0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0 |
| **1** | 0.0 | 1.057999e-07 | 0.000000e+00 | 0 | 0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0 |
| **2** | 0.0 | 0.000000e+00 | 0.000000e+00 | 0 | 0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0 |
| **3** | 0.0 | 1.681203e-07 | 6.223962e-06 | 0 | 0 | 0.0 | 0.0 | 0.0 | 1 | 0.0 | 0 |
| **4** | 0.0 | 1.442067e-07 | 3.206260e-07 | 0 | 0 | 0.0 | 0.0 | 0.0 | 1 | 0.0 | 0 |

***Dataset after preprocessing***



***Heatmap - Correlation Matrix***

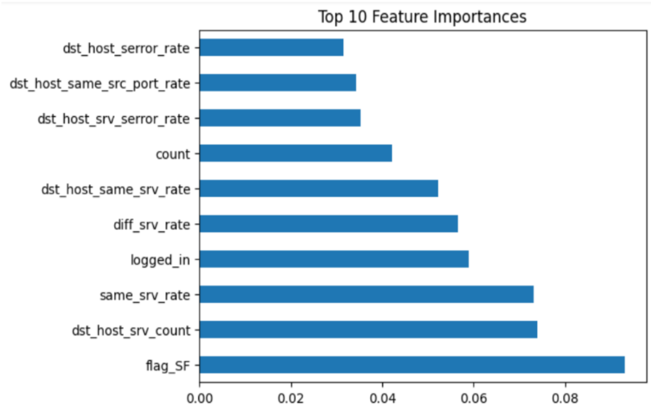
# OUTPUT SCREENSHOTS (cont.)

Attributes with high positive correlation with label

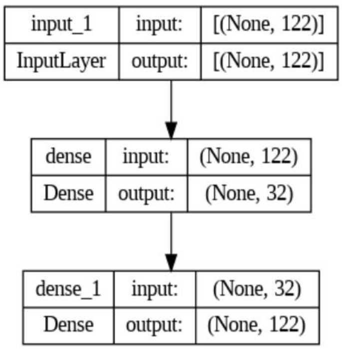
- flag_SF = 0.727673
- same_srv_rate = 0.708911
- dst_host_srv_count = 0.692577
- dst_host_same_srv_rate = 0.667624
- logged_in = 0.664117
- service_http = 0.567600

Attributes with high negative correlation with label

- count = -0.524108
- flag_S0 = -0.585611
- srv_serror_rate = -0.586636
- serror_rate = -0.588474
- dst_host_serror_rate = -0.589936
- dst_host_srv_serror_rate = -0.593690




# OUTPUT SCREENSHOTS (cont.)




***AutoEncoder model***

```
normal count : 9191
non normal count : 809
```

***Test with normal traffic data***

```
anomalous count : 59669
non anomalous count : 11794
```

***Test with anomalous traffic data***

# NEXT PHASE OF WORK

1. Determine suitable clustering technique and implement the clustering module.
2. Build a deep autoencoder model for each cluster.
3. Evaluate the performance of the system and fine tune the parameters.

# REFERENCES


1. W. T. Lunardi, M. A. Lopez and J. -P. Giacalone, "ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1305-1318, June 2023, doi: 10.1109/TNSM.2022.3229706. keywords: {Anomaly detection;Training;Telecommunication traffic;Generative adversarial networks;Data models;Generators;Deep learning;Unsupervised anomaly detection;autoencoder;generative adversarial networks;automatic feature extraction;deep learning;cybersecurity},
2. N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792. keywords: {Machine learning;Intrusion detection;Anomaly detection;Training data;Communication networks;Monitoring;Deep learning;anomaly detection;auto-encoders;KDD;network security},
3. C. Aytekin, X. Ni, F. Cricri and E. Aksu, "Clustering and Unsupervised Anomaly Detection with l2 Normalized Deep Auto-Encoder Representations," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-6, doi: 10.1109/IJCNN.2018.8489068. keywords: {Training;Anomaly detection;Neural networks;Image reconstruction;Encoding;Clustering algorithms;Clustering methods},
4. I. Ursul and A. Pereymybida, "Unsupervised Detection of Anomalous Running Patterns Using Cluster Analysis," 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 2023, pp. 148-152, doi: 10.1109/ELIT61488.2023.10310751. keywords: {Performance evaluation;Wearable computers;Scalability;Clustering algorithms;Network intrusion detection;Optics;Telemetry;clusters;anomaly detection;unsupervised learning},
5. S. S. Khan and A. B. Mailewa, "Detecting Network Transmission Anomalies using Autoencoders-SVM Neural Network on Multi-class NSL-KDD Dataset," 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2023, pp. 0835-0843, doi: 10.1109/CCWC57344.2023.10099056. keywords: {Support vector machines;Training;Computational modeling;Neural networks;Predictive models;Feature extraction;Security;NSL-KDD;Network Security;Intrusion Detection System;Deep Autoencoders;Anomaly Detection;Support Vector Machine (SVM);t-SNE;Deep Learning},

# REFERENCES (cont.)


6. W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei and F. Sabrina, "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset," in IEEE Access, vol. 9, pp. 140136-140146, 2021, doi: 10.1109/ACCESS.2021.3116612. keywords: {Anomaly detection;Data models;Training;Network security;Mathematical models;Encoding;Task analysis;Network security;intrusion detection systems;network-based IDSs;anomaly detection;NSL-KDD;artificial intelligence;machine learning;deep learning;autoencoders;unsupervised learning},

7. C. Maudoux and S. Boumerdassi, "Network Anomalies Detection by Unsupervised Activity Deviations Extraction," 2022 Global Information Infrastructure and Networking Symposium (GIIS), Argostoli, Greece, 2022, pp. 1-5, doi: 10.1109/GIIS56506.2022.9937022. keywords: {Clustering algorithms;Organizations;Computer crime;Anomaly detection;Sports;Geohash;anomalies detection;activity deviation;unsupervised machine learning;networks;aggregation;security},

8. C. Park, J. Lee, Y. Kim, J. -G. Park, H. Kim and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," in IEEE Internet of Things Journal, vol. 10, no. 3, pp. 2330-2345, 1 Feb.1, 2023, doi: 10.1109/JIOT.2022.3211346.

9. H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari and J. Qadir, "Robust Enhancement of Intrusion Detection Systems Using Deep Reinforcement Learning and Stochastic Game," in IEEE Transactions on Vehicular Technology, vol. 71, no. 10, pp. 11089-11102, Oct. 2022, doi: 10.1109/TVT.2022.3186834

10. N. Ravi and S. M. Shalinie, "Semisupervised-Learning-Based Security to Detect and Mitigate Intrusions in IoT Network," in IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11041-11052, Nov. 2020, doi: 10.1109/JIOT.2020.2993410.

11. M. A. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," in IEEE Transactions on Computers, vol. 65, no. 10, pp. 2986-2998, 1 Oct. 2016, doi: 10.1109/TC.2016.2519914.