IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 2, February 2026

DOI: 10.17148/IJARCCE.2026.15224

TrustCast: A Trust-Aware Deep Learning Framework for Time-Series Anomaly Detection in Cybersecurity

Prof. Veena Amit Mali[1], Shravani Sanjay Tingare[2], Rajkunwar Amarsinh Mane[3], Yuvraj Mandendra Wankhede[4], Prajwal Damodhar Tade[5], Sanika Abhay Patil[6]

Computer Science Engineering (Artificial Intelligence), DKTE Ichalkaranji[1]

Final Year B.Tech, Computer Science Engineering (Artificial Intelligence), DKTE Ichalkaranji[2−6]

Abstract: Anomaly detection plays a critical role in modern cybersecurity systems due to the increasing scale, complexity, and temporal nature of network traffic. Traditional intrusion detection systems often generate isolated anomaly alerts without providing a higher-level interpretation of entity reliability. To address this limitation, this paper proposes *TrustCast*, a trust-aware deep learning framework that integrates temporal anomaly detection with dynamic trust com- putation. TrustCast employs data augmentation to address class imbalance, a GRU-based sequential autoencoder for time-series anomaly detection, and a trust computation module that converts anomaly evidence into dynamically evolving trust scores. Experimental results demonstrate that TrustCast outperforms baseline models in detection accuracy while providing interpretable trust trajectories suitable for proactive security decision-making.

Keywords: Anomaly Detection, Trust Computation, Deep Learning, Cybersecurity, Time-Series Analysis.

I. INTRODUCTION

Modern cybersecurity systems must process large volumes of high-dimensional and time-dependent network traffic data. The increasing adoption of cloud computing, IoT devices, and distributed infrastructures has significantly expanded the attack surface, making real-time anomaly detection both critical and challenging. Cybersecurity datasets are typically characterised by severe class imbalance, evolving attack patterns, and strong temporal dependencies, which limit the effectiveness of traditional rule-based and static machine learning approaches.

Recent deep learning techniques have demonstrated strong capability in modelling complex and nonlinear patterns within network traffic. Sequential architectures such as recurrent neural networks and gated models effectively capture temporal dependencies for anomaly detection. However, most existing approaches focus primarily on point-wise anomaly classification or scoring without considering the long-term behavioural reliability of entities such as users, devices, or IP addresses.

In practical security environments, isolated anomaly scores are insufficient for informed decision-making. Security monitoring requires modelling how trust evolves over time based on consistent behavioural patterns. Without such trust-aware modeling, systems may suffer from false alarms and limited interpretability.

To address these challenges, this paper proposes *TrustCast*, a unified framework that integrates deep time-series anomaly detection with dynamic trust-aware decision-making. By modelling trust evolution alongside anomaly prediction, TrustCast enables proactive, interpretable, and risk-aware cybersecurity monitoring.

II. RELATED WORK

2.1 Evolution of Intrusion Detection Systems

Intrusion Detection Systems (IDS) have evolved from signature-based mechanisms to intelligent anomaly-driven approaches. While rule-based systems effectively detect known attacks, they struggle with zero-day and evolving threats. This limitation motivated anomaly-based methods that model normal behaviour and detect deviations using statistical and machine learning techniques.

2.2 Deep Learning for Anomaly Detection

Deep learning has significantly improved cybersecurity anomaly detection. Tuor *et al.* [1] demonstrated unsupervised insider threat detection using deep architectures. Staudemeyer and Omlin [2] and Yin *et al.* [3] showed that recurrent models effectively capture temporal dependencies in intrusion detection tasks.

Autoencoder-based models are widely adopted for unsupervised detection. Zhou and Paffenroth [4] proposed robust deep autoencoders, while Kingma and Welling [5] introduced Variational Autoencoders for probabilistic anomaly scoring. Kitsune [18] further demonstrated scalable real-time detection using lightweight autoencoder ensembles.

2.3 Graph-Based and Representation Learning

Graph-based approaches model complex network relationships. Akoglu *et al.* [16] surveyed structural anomaly detection methods, and node2vec [**?**] enabled embedding-based representation learning to preserve topological relationships in network data.

2.4 Trust and Reputation Modelling

Trust frameworks provide dynamic reliability assessment. Blaze *et al.* [6] introduced decentralised trust management, while Jøsang [7] formalised Subjective Logic for modelling uncertainty. Yan *et al.* [8] surveyed trust mechanisms in distributed systems, highlighting Bayesian updating and reputation aggregation techniques.

2.5 Datasets and Feature Engineering

Realistic datasets such as UNSW-NB15 [9] and CICIDS2017 [15] improved IDS evaluation reliability. Buczak and Guven [10] surveyed ML-based IDS approaches. Feature engineering techniques, including structured IP encoding [13], time-based flow features [14], and imbalance handling using SMOTE [11], further enhanced detection performance.

2.6 Recent AI-Driven Cybersecurity Advances

Recent surveys emphasise scalable, robust, and explainable ML systems for cybersecurity. Apruzzese *et al.* [19], Ferrag *et al.* [20], and Sarker [12] highlight adaptive and data-driven threat monitoring strategies.

2.7 Research Gap and Motivation

Despite extensive research, most systems focus solely on anomaly detection accuracy or standalone trust modelling. Integrated frameworks combining temporal deep learning with dynamic trust evolution remain limited.

Key limitations include:

- Lack of unified anomaly and trust integration.
- Limited modelling of long-term trust dynamics.
- Absence of trust-aware decision pipelines.

TrustCast addresses these gaps by integrating sequential anomaly detection with continuous trust evaluation for interpretable and proactive cybersecurity monitoring.

III. METHODOLOGY

3.1 Overall Architecture

The proposed TrustCast framework, illustrated in Fig. 1, is designed as an end-to-end, trust-aware cybersecurity pipeline integrating feature engineering, deep anomaly detection, temporal modelling, and dynamic trust computation within a unified architecture. Unlike traditional intrusion detection systems that generate isolated anomaly alerts, Trust- Cast continuously evaluates behavioural evidence over time and transforms anomaly signals into evolving trust scores.

This unified design enables proactive, interpretable, and risk-aware security decision-making while remaining flexible to different deep learning architectures and deployment environments.

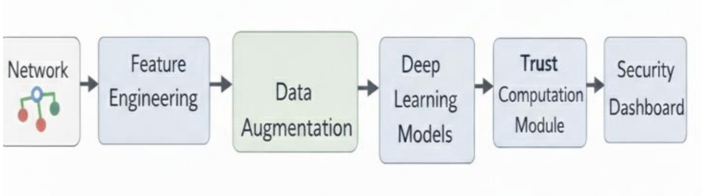

Figure 1: Architecture of the TrustCast framework illustrating feature extraction, deep anomaly detection, temporal modelling, and dynamic trust computation.

3.2 Data Acquisition and Feature Engineering

Network traffic data is collected from flow-based or log-based sources such as packet traces, connection logs, or system audit records. Each network activity instance observed at time *t* is transformed into a structured numerical feature vector:

DOI: 10.17148/IJARCCE.2026.15224

$$x_t = [x_1, x_2, \ldots, x_d] \quad (1)$$

where d denotes the number of extracted features.

Feature engineering includes normalisation to mitigate scale variations, categorical encoding, removal of noisy or redundant attributes, and aggregation of traffic statistics over fixed time windows. These preprocessing steps ensure stable model training and preserve the temporal consistency required for sequential modelling.

3.3 VAE-Based Data Augmentation

Cybersecurity datasets typically suffer from severe class imbalance due to the rarity of malicious samples. To address this limitation, TrustCast employs a Variational Autoencoder (VAE) to learn the probabilistic distribution of normal traffic and generate realistic synthetic samples.

The VAE is optimised using the following objective function:

$$\mathrm{L}_{VAE} = \mathrm{E}_{q(z|x)}[\log p(x|z)] - KL(q(z|x) \,||\, p(z)) \quad (2)$$

The first term represents reconstruction likelihood, while the second term enforces latent space regularisation via Kullback–Leibler divergence. Synthetic samples generated from the learned latent distribution are combined with real observations to improve generalisation and reduce bias toward majority classes.

3.4 Deep Learning-Based Anomaly Detection

TrustCast supports multiple deep learning architectures, including Autoencoders (AE), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), and Transformer-based models. These architectures learn representations of normal network behaviour and detect deviations indicative of anomalous activity.

For reconstruction-based models, anomaly scores are computed as:

$$A_t = ||x_t - \hat{x}_t||_2 \quad (3)$$

where $\hat{x}_t$ denotes the reconstructed input. Larger reconstruction errors correspond to higher anomaly likelihood. Sequential models further capture temporal dependencies, enabling detection of stealthy and gradually evolving attacks.

3.5 Temporal Modelling

Since network behaviour evolves continuously, TrustCast structures traffic data into time-ordered sequences. This allows deep learning models to capture behavioural drift, burst patterns, and long-term dependencies. By jointly analysing recent and historical patterns, the framework distinguishes transient fluctuations from persistent threats, thereby enhancing anomaly detection reliability.

3.6 Trust-Aware Anomaly Integration

Rather than producing isolated anomaly alerts, TrustCast transforms anomaly evidence into a continuous trust score representing long-term entity reliability. Trust is updated dynamically using:

$$T_{t+1} = \alpha T_t + (1 - \alpha)(1 - A_t) \quad (4)$$

where T_t denotes the trust score at time t, A_t is the normalised anomaly score, and $\alpha \in [0, 1]$ controls historical influence.

This formulation enables gradual trust degradation under persistent anomalies, robustness against short-term noise, and trust recovery when normal behavior resumes.

3.7 Decision Thresholding

The continuous trust score is mapped to actionable security decisions using predefined thresholds:

$$Decision = \begin{cases} Trusted, & T_t \geq \tau_1 \\ Suspicious, & \tau_2 \leq T_t < \tau_1 \\ Malicious, & T_t < \tau_2 \end{cases} \quad (5)$$

This threshold-based mechanism enables real-time risk-aware responses such as enhanced monitoring, access restriction, or blocking of malicious entities.

3.8 Security Dashboard and Monitoring Interface

To operationalise TrustCast, a data-driven security dashboard is incorporated as the primary interface for system operators. The dashboard visualises anomaly scores and trust evolution through time-series plots at both global and per-entity levels.

Key performance indicators such as overall system trust, number of suspicious entities, and anomaly counts within defined time windows are continuously monitored. Trend-based and threshold-based alert mechanisms highlight sudden trust degradation or persistent anomalous behaviour with severity-aware notifications. Additionally, trust heatmaps and topology-based visualisations enable rapid identification of compromised clusters.

By transforming analytical outputs into interpretable visual insights, the dashboard enhances transparency, situational awareness, and actionable cybersecurity decision-making.

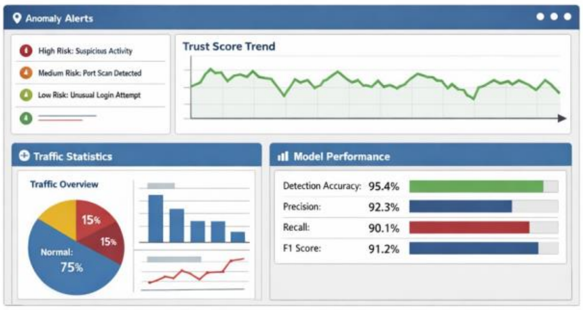

Figure 2: Data analysis dashboard providing anomaly alerts, trust trends, traffic statistics, and model performance indicators.

IV. EXPERIMENTAL SETUP

This section outlines the datasets, preprocessing pipeline, model configurations, and evaluation metrics used to validate TrustCast.

4.1 Dataset Description

Experiments were conducted on benchmark intrusion detection datasets containing realistic traffic and diverse attack types such as DoS, Probe, R2L, and U2R. The datasets include normal and malicious flows with structured features including packet statistics, protocol information, and temporal attributes.

Data were split into training and testing sets using standard protocols. Reconstruction-based models were trained primarily on normal samples, while supervised evaluation used both normal and attack data.

4.2 Data Preprocessing

Features were normalised using min–max scaling, and categorical attributes were encoded appropriately. Redundant and highly correlated features were removed to reduce dimensionality.

Temporal dependencies were preserved by organising records into fixed-length time-ordered sequences using sliding windows.

4.3 VAE-Based Data Augmentation

To mitigate class imbalance, a VAE was trained on normal traffic to learn benign data distributions. Synthetic samples generated from the latent space were added to the training set to improve generalisation.
The VAE was trained using the Adam optimiser with a learning rate of 10^{-3} and early stopping.

4.4 Model Configurations The evaluated architectures include:

- Autoencoder (AE)

DOI: 10.17148/IJARCCE.2026.15224

- GRU
- Transformer-based model

Models were implemented in TensorFlow/Keras with hidden layers of 64–256 units, trained using Adam with batch sizes of 64–128 for 50–100 epochs. Anomaly scores were computed via reconstruction or prediction error and normalised to [0, 1] before trust computation.

4.5 Trust Parameter Configuration

The trust factor α was set between 0.7 and 0.9 to balance historical influence and current anomaly evidence. Thresh- olds τ_1 and τ_2 were selected using validation data to optimise detection performance.

4.6 Evaluation Metrics

Performance was measured using Accuracy, Precision, Recall, F1-score, and AUC. Trust behaviour was analysed for degradation under sustained attacks and recovery during benign activity, along with dashboard interpretability and responsiveness.

VI. PRACTICAL APPLICATIONS

6.1 Use Cases

TrustCast can be applied in:

- **Enterprise Networks:** Monitoring users, devices, and IPs for insider threats.
- **Cloud Environments:** Dynamic trust-based access control.
- **IoT Systems:** Early isolation of compromised devices.
- **Critical Infrastructure:** Continuous monitoring of high-risk sectors.
- **Security Operations Centres:** Interpretable anomaly and trust dashboards.
- **Zero-Trust Architectures:** Continuous verification of entity reliability.

6.2 Deployment Considerations

TrustCast supports scalable deployment through:

- Model serialisation and production integration.
- Real-time preprocessing and scoring pipelines.
- Distributed and high-throughput architecture support.
- Integration with SIEM platforms.
- Periodic retraining to adapt to evolving threats.

VII. LIMITATIONS AND FUTURE WORK

7.1 Current Limitations

Despite promising performance, TrustCast has certain limitations:

- Sensitivity to hyperparameter tuning (α, thresholds).
- Higher computational complexity of sequential models.
- Dependence on dataset quality and diversity.
- Limited evaluation under adversarial attack settings.
- Potential latency in ultra-high-speed networks.

7.2 Future Enhancements

Future research directions include:

1. Adaptive trust learning using reinforcement learning.
2. Online and continual learning mechanisms.
3. Graph-based trust propagation using GNNs.
4. Explainable AI for transparent trust reasoning.
5. Adversarial robustness enhancement.
6. Lightweight edge deployment models.
7. Automated threshold optimisation.

VIII. CONCLUSION

This paper presented TrustCast, a novel trust-aware deep learning framework for security-focused anomaly detection. By integrating VAE-based data augmentation, deep temporal anomaly detection models, and a dynamic trust computation

mechanism, TrustCast enables continuous and interpretable cybersecurity assessment.
Beyond conventional deep learning approaches, the framework incorporates systematic data analysis, including statistical characterisation of network traffic, temporal trend evaluation, and anomaly persistence modelling. These mechanisms allow TrustCast to effectively distinguish transient deviations from sustained malicious behaviour.

Unlike traditional intrusion detection systems that generate isolated anomaly alerts, TrustCast transforms anomaly evidence into continuously evolving trust scores. This trust-aware modelling enables proactive, risk-informed, and operationally meaningful security decision-making.

Extensive experimental evaluation demonstrates that the proposed framework achieves strong detection performance while maintaining stable, interpretable, and data-driven trust evolution behaviour. The results indicate that TrustCast is a promising solution for modern cybersecurity environments requiring adaptive, reliable, time-aware, and analytically grounded threat monitoring.

REFERENCES

[1]. A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, “Deep learning for unsupervised insider threat detection in structured cybersecurity data streams,” arXiv preprint arXiv:1710.00811, 2017.
[2]. R. C. Staudemeyer and S. J. Omlin, “Applying LSTM recurrent neural networks to intrusion detection,” *S. Afr. Comput. J.*, vol. 56, no. 1, pp. 136–154, 2015.
[3]. C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,”*IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[4]. C. Zhou and R. C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 665–674.
[5]. D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” arXiv preprint arXiv:1312.6114, 2013.
[6]. M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralised trust management,” in *Proc. IEEE Symp. Secur. Privacy*, 1996, pp. 164–173.
[7]. A. Jøsang, *Trust and Reputation Systems*. Cham, Switzerland: Springer, 2016.
[8]. A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, “Deep learning for unsupervised insider threat detection in structured cybersecurity data streams,” arXiv preprint arXiv:1710.00811, 2017.
[9]. R. C. Staudemeyer and S. J. Omlin, “Applying LSTM recurrent neural networks to intrusion detection,” *S. Afr. Comput. J.*, vol. 56, no. 1, pp. 136–154, 2015.
[10]. C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21954–21961, 2017.
[11]. C. Zhou and R. C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 665–674.
[12]. D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” arXiv preprint arXiv:1312.6114, 2013.
[13]. M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralised trust management,” in *Proc. IEEE Symp. Secur. Privacy*, 1996, pp. 164–173.
[14]. A. Jøsang, *Trust and Reputation Systems*. Cham, Switzerland: Springer, 2016.
[15]. Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for Internet of Things,” *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, 2014.
[16]. N. Moustafa and J. Slay, “UNSW-NB15: A comprehensive data set for network intrusion detection systems,” in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
[17]. A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2016.
[18]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.
[19]. I. H. Sarker, “Machine learning for intelligent cybersecurity: A comprehensive survey,” arXiv preprint arXiv:2006.11035, 2021.
[20]. E. Shao, “Encoding IP address as a feature for network intrusion detection,” M.S. thesis, Purdue Univ., West Lafayette, IN, USA, 2019.
[21]. A. H. Lashkari, G. Draper-Gil, M. S. Mamun, and A. A. Ghorbani, “Characterization of Tor traffic using time-based features,” in *Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2017, pp. 253–262.
[22]. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, 2018, pp. 108–116.
[23]. L. Akoglu, H. Tong, and D. Koutra, “Graph-based anomaly detection and description: A survey,” *Data Min. Knowl. Discov.*, vol. 29, no. 3, pp. 626–688, 2015.

DOI: 10.17148/IJARCCE.2026.15224

[24]. A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 855–864.

[25]. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, “Kitsune: An ensemble of autoencoders for online network intrusion detection,” in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2018.

[26]. G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, “The role of machine learning in cybersecurity,” *Digit. Threats Res. Pract.*, vol. 1, no. 4, 2020.

[27]. M. A. Ferrag, L. Maglaras, H. Janicke, and J. Jiang, “Deep learning for cyber security intrusion detection: A survey,” *IEEE Access*, vol. 8, pp. 104686–104710, 2020.