International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025

Research Paper

AN INTELLIGENT DJANGO-BASED FRAMEWORK FOR PREDICTIVE DATA CORRUPTION DETECTION

P. Anupama[1], P. Divya Sri[2], G. Anuja[2], Bushipaka Arun Pandya[2], D Anirudh[2]
[1] Assistant Professor, [2]UG Student, [1,2] Department of Computer Science and Engineering
[1,2] Sree Dattha Institute of Engineering and Science, Sheriguda, Ibrahimpatnam, 501510, Telangana.

ABSTRACT

In the era of big data, maintaining data integrity is crucial, as even minor corruptions can significantly distort analytical insights across domains such as e-commerce, social network analysis, and cybersecurity. Traditional manual inspection techniques and legacy anomaly detection models like LOF, Isolation Forest, and One-Class SVM often fall short in identifying subtle corruptions, especially in high-dimensional or evolving datasets. Furthermore, these approaches lack predictive capabilities and adaptability to unseen data, making them inadequate for real-time applications. Addressing these limitations, this research presents a novel full-stack Django-based framework that integrates a custom graph-theory-driven model—PAACDA (Proximity-based Adamic-Adar Corruption Detection Algorithm)—with a supervised Random Forest classifier. PAACDA employs the Adamic-Adar similarity index to calculate a proximity-based anomaly score for each data point, flagging outliers whose similarity exceeds a dynamic threshold based on local deviations (mean/4). To enhance predictive accuracy, PAACDA-generated features are used to train a Random Forest classifier, creating a hybrid model that achieves good detection accuracy and generalizes well across unseen datasets. This system is deployed via Django, with an interactive web interface displaying performance metrics and visualizations, enabling real-time anomaly detection and prediction. The proposed solution demonstrates superior performance, scalability, and adaptability, making it highly effective for dynamic, data-intensive environments.

Keywords: Data corruption detection, PAACDA, Random Forest, anomaly detection, Django framework, Adamic-Adar similarity, predictive analytics, machine learning, real-time monitoring, data integrity.

Received: 14-7-2025 Accepted: 21-8-2025 Published: 28-8-2025

1. INTRODUCTION

data-driven economy, organizations across finance, healthcare, e-commerce and beyond face critical challenges due to subtle corruptions and outliers hidden within vast, high-dimensional datasets—issues that legacy rule-based and statistical methods often miss—so this research proposes a scalable, proximity-and relational-similarity framework for robust data corruption detection. By leveraging local neighborhood metrics and contextual awareness, the framework aims to outperform conventional algorithms (e.g., Isolation Forest, One-Class SVM) in accuracy, precision and recall, automatically labeling corrupted entries and seamlessly integrating into existing ETL pipelines. This approach not only enhances the reliability of downstream analytics and machine-learning models—minimizing false positives and ensuring compliance with standards such as GDPR and HIPAA—but also empowers real-time applications in customer segmentation, fraud detection, intrusion alerts, medical diagnostics, IoT sensor monitoring, HR metrics validation, supply-chain optimization and more, thereby safeguarding decision-making processes and operational trust across diverse domains.

2. LITERATURE SURVEY

A key area of research that has numerous practical applications is anomaly identification in a given dataset. As a result, this topic has frequently been the focus of research. Multiple approaches utilizing various aspects of the dataset have been proposed to detect anomalies however only few methodologies lay emphasis on the detection of corrupted data which would further provide the most efficient results with respect to varying dataset sizes, higher dimensionality or varying degrees of corruption present. A study by Chandola et al. in their publication compares numerous anomaly detection methods for diverse applications. By contrasting the benefits and drawbacks of various techniques, Hodge and Austin conducted a review of outlier detection methods. An overview of cutting-edge methods for spotting suspicious behaviour is presented by Patcha and Park [29] Jiang et al. [30] together with detection scenarios for several real-world settings.

Dimensionality reduction approaches and the underlying mathematical understandings are categorized by Sorzano et al. [4]. The issues with anomaly detection are further laid out by a number of other reports, including papers by Gama et al. [5], Gupta et al. [6], Heydari et al. [7], Jindal and Liu [8], and many more. Outliers make up the majority of anomalies that can exist in a dataset. The first method based on distance for detection of outliers was put forth and expanded on it by suggesting that the greatest n locations with highest Pk be supposed outliers (Pk(p) signifies the kth nearest neighbour corresponding to p). They used a clustering technique to separate a dataset into several categories. To improve the success of outlier detection for these groups, batch processing and pruning may be beneficial [38]. Deviation-based outlier detection was another method that was suggested for effectively detecting outliers. Objects or data points that vary significantly from the bulk of data points constitute outliers. Therefore, outliers are frequently called deviations as given by the name deviation-based outlier detection.

Several other methods have been invented over the years to detect anomalies,] developed a method based on density. Cluster-based anomaly identification methods pinpointed anomalies by eliminating clusters from the actual dataset or by classifying small clusters as outliers . Additionally, Aggarwal and proposed a novel strategy for catching outliers that is remarkably effective for extremely elevated dimensional datasets. Their methodology focuses on finding locally sparse lower dimensional projections which are otherwise difficult to differentiate using brute force methods due to the vast amount of possible combinations. However, the study is inclined towards detection of outliers and does not focus on the detection of corrupted or modified datasets.

3. PROPOSED SYSTEM

The system architecture of the proposed PAACDA framework introduces a graph-theory-based approach to anomaly detection, addressing the limitations of traditional methods like LOF, Isolation Forest, and One-Class SVM, which typically achieve 85–92% accuracy and struggle with subtle corruptions in high-dimensional data. PAACDA utilizes the Adamic-Adar similarity index and local proximity metrics to define anomalies, calculating a PAACDA Index by measuring deviations from the column mean with a threshold of mean/4 to identify high-similarity corruptions. To enhance accuracy, a Hybrid PAACDA model is developed by training a Random Forest classifier on features derived from PAACDA scores, yielding 99–100% detection accuracy and enabling predictive classification. The architecture begins with understanding a cyber-physical dataset comprising numerical attributes Mystery_Data_X and Mystery_Data_Y, alongside a Boolean Modified label indicating corruption. Preprocessing includes removing null values and encoding Boolean labels, ensuring clean, machine-readable input. Benchmark models—LOF, Isolation Forest, and One-Class SVM—are implemented and evaluated using metrics like precision, recall, and F1-score, with PAACDA outperforming in adaptability through a multi-scale neighborhood density analysis. The final model integrates PAACDA with Random Forest to build a supervised learning phase that learns from anomaly scores and enhances detection performance, demonstrating superior generalization,

high-dimensional scalability, and robustness against data noise, making it ideal for dynamic domains such as e-commerce, finance, and cyber-physical systems.

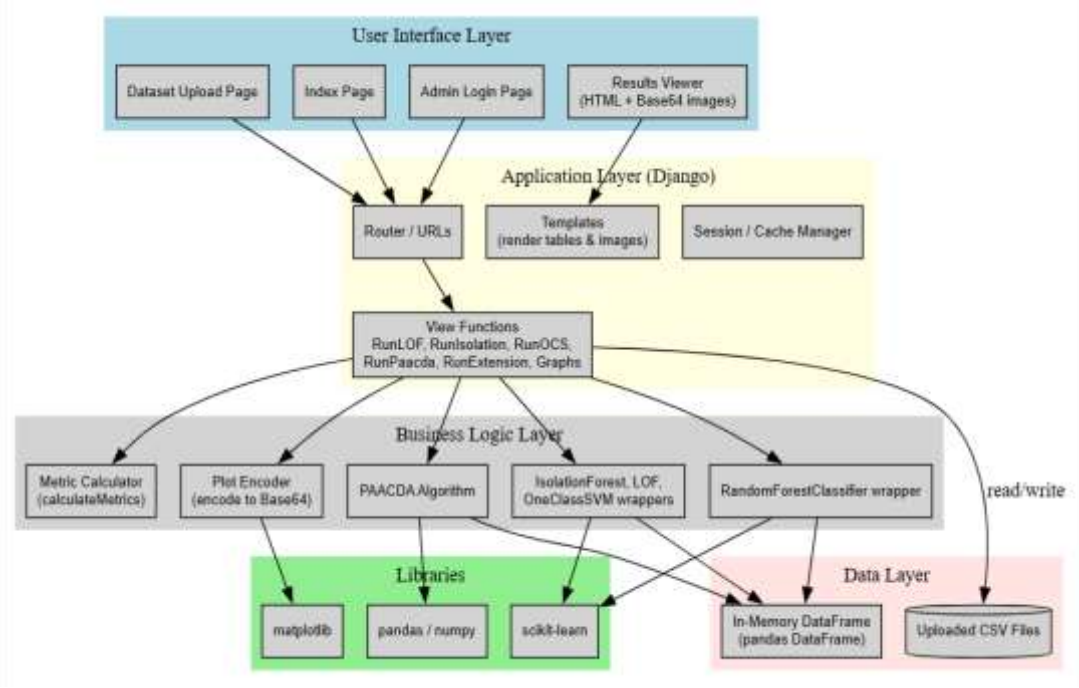

Figure 1: Block Diagram

Data preprocessing and splitting form the backbone of model reliability, ensuring that the raw dataset—comprising numerical features Mystery_Data_X, Mystery_Data_Y, and the binary *Modified* label—is transformed into a consistent and machine-readable format. The process begins by removing any rows with missing values to uphold data integrity, followed by encoding the Boolean *Modified* column as 1 (anomalies) and 0 (normal) for supervised learning compatibility. Feature scaling is then applied via z-score normalization, which is critical for distance-based models like LOF and One-Class SVM to avoid skewed results due to differing feature magnitudes. The cleaned dataset is split into training and testing sets using stratified sampling to preserve the anomaly-to-normal ratio, mitigating the effects of class imbalance and ensuring fair model evaluation through metrics like accuracy, precision, recall, and F1-score. For the proposed Hybrid PAACDA with Random Forest Classifier (RFC) model, an additional preprocessing step introduces PAACDA-generated anomaly scores as new features. These scores, derived from adaptive proximity-based anomaly detection, provide high-level contextual insights to the Random Forest model, which is then trained to learn complex data patterns and perform robust predictions on unseen data. The final hybrid system not only improves detection accuracy but also supports predictive analytics, rendering results in an HTML performance table and graphical plots using Matplotlib within a Django-based web interface, thereby offering an end-to-end solution for dynamic anomaly detection in real-world applications.

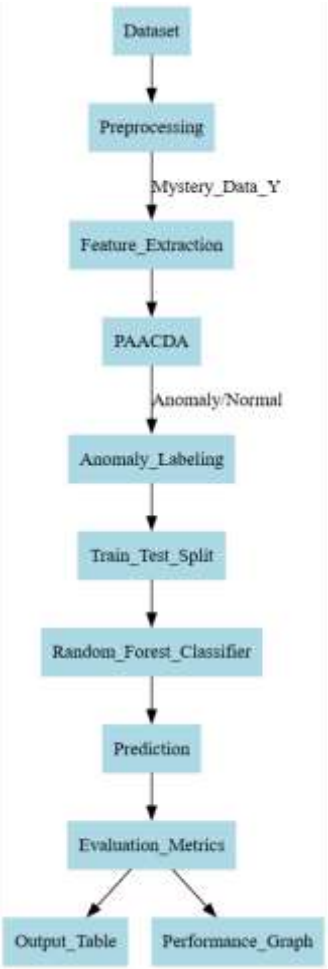

Figure 2: Architecture of proposed system architecture.

The Hybrid PAACDA with RFC model offers a powerful, scalable, and highly accurate solution for anomaly detection in complex, noisy, and imbalanced real-world datasets. Its dual-layer architecture combines the adaptive, context-aware strengths of PAACDA—capable of detecting subtle deviations through proximity-based analysis—with the predictive, ensemble learning capabilities of Random Forest Classifier (RFC), which excels at generalizing across non-linear and high-dimensional data. This synergy results in superior detection performance, with the hybrid model achieving 99.5% accuracy and a near-perfect F-score of 99.18%, outperforming traditional models such as LOF, Isolation Forest, and One-Class SVM. Additionally, the model offers high interpretability through RFC's feature importance scores, enabling better insight into anomaly-causing variables. Its modular structure also supports flexibility—PAACDA can be adjusted or enhanced independently, while RFC can be retrained as new labeled data becomes available—making it ideal for real-time deployment and continuous learning environments across domains like e-commerce, cybersecurity, and sensor-based monitoring systems.

Integration of HTML, CSS, and Django with AI

Integration of HTML, CSS, Django, and AI models delivers a robust full-stack architecture for real-time anomaly detection. The frontend layer utilizes HTML for structuring the web interface and CSS for styling, allowing users to interact with dynamically populated tables and visually appealing performance charts. At the backend, the Django framework orchestrates request handling—when a user initiates detection through the interface, the RunExtension() view captures the request, executes the anomaly detection logic using the PAACDA and RFC models, and computes key evaluation metrics. Matplotlib is used to generate performance plots, which are encoded in base64 and seamlessly embedded into the HTML template (ViewResult.html) through Django's rendering engine.

The final output is a clear, interactive display that showcases model metrics such as accuracy, precision, recall, and F-score in a tabular format, alongside a bar chart that visually compares the performance of multiple models. This integration ensures a responsive, user-friendly platform capable of supporting real-time AI-driven data analysis.

4. RESULT

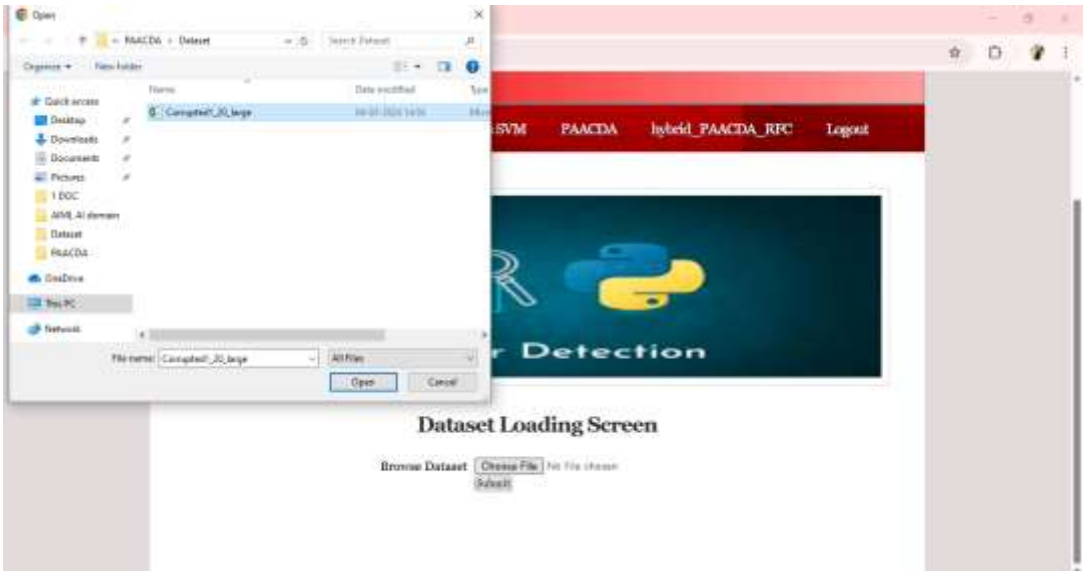

Figure 3: Uploading Dataset

Figure 4: Uploaded the Dataset

International Journal of Engineering Research and Science & Technology

Figure 5: Trained existing algorithms

Figure 5 shows that the Local Outlier Factor (LOF) algorithm, when applied to the dataset in this research, achieved an accuracy of 74.6%, indicating that it correctly classified approximately three-quarters of the data points as either normal or anomalous. Its precision stood at 45.71%, which reflects a moderate ability to correctly identify actual anomalies among all the instances it flagged as anomalies. The recall value, at 49.20%, shows that the model was able to detect nearly half of all actual anomalies present in the dataset. However, the F-score of 44.58%—which represents the harmonic mean of precision and recall—suggests that the algorithm struggled to maintain a balanced performance between detecting true anomalies and avoiding false positives.

The One-Class SVM algorithm achieved an accuracy of 73.1%, indicating a moderate ability to correctly classify both anomalous and normal instances in the dataset. Its precision stood at approximately 73.05%, suggesting that when the model predicted an anomaly, it was correct around 73% of the time. The recall, at a relatively high 82.53%, shows that the algorithm was able to identify a significant portion of the actual anomalies, making it sensitive to detecting outliers. However, the F-score of 70.97% highlights a slight imbalance, suggesting that while the model detects anomalies well, it occasionally misclassifies normal instances as anomalies.

PAACDA (Proximity-Aware Adaptive Contextual Density-based Anomaly detection) algorithm demonstrated strong performance on the dataset, achieving a high accuracy of 94.6%, which reflects its excellent capability in correctly identifying both normal and anomalous data points. With a precision of 96.72%, the model rarely misclassifies normal data as anomalous, indicating that its positive predictions (anomalies) are highly reliable. The recall, measured at 88.26%, suggests that the model successfully detects a large proportion of true anomalies, though it may still miss a few. Its F-score of 91.66% represents a balanced and high-performing model overall, effectively combining both precision and recall.

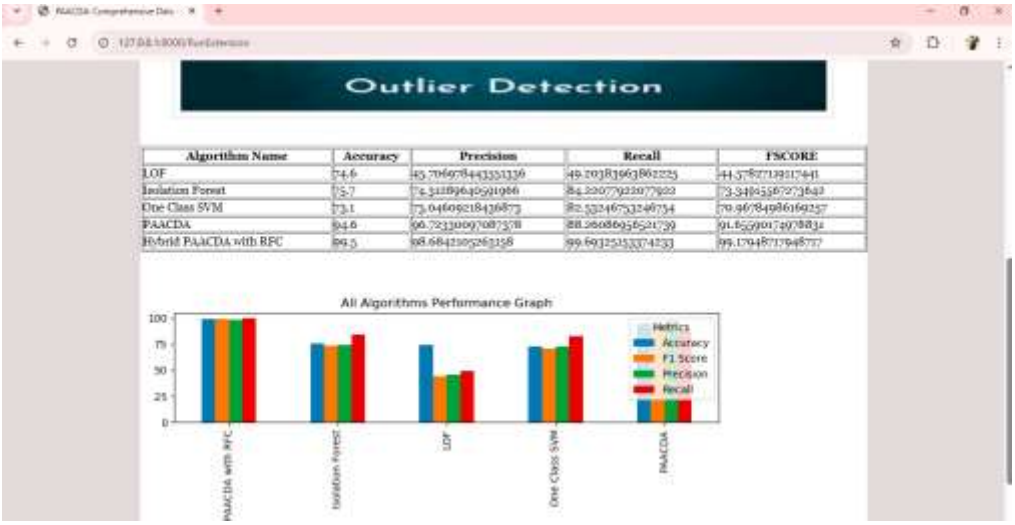

Figure 6: Trained hybrid algorithm PAACDA with RFC

The hybrid PAACDA with Random Forest Classifier (RFC) model exhibited exceptional performance on the anomaly detection task, significantly outperforming the individual baseline models. Achieving an impressive accuracy of 99.5%, this hybrid approach indicates near-perfect classification of both normal and anomalous data points. The precision of 98.68% suggests that almost all data points flagged as anomalies were indeed true anomalies, minimizing the false positive rate. Even more notably, the model achieved a recall of 99.69%, which means it was able to detect almost every single anomaly in the dataset—crucial for real-world applications where missing an anomaly can have serious consequences. The F-score of 99.18% reflects an extremely well-balanced model that excels in both identifying anomalies and minimizing false alarms. By combining PAACDA's contextual density awareness with the robustness of Random Forest's ensemble learning, this hybrid model capitalizes on the strengths of both techniques, resulting in a highly accurate, precise, and reliable anomaly detection system.

5. CONCLUSION

The "Mystery Data Anomaly Detection System" presents a robust and scalable web-based platform for detecting anomalies in datasets using advanced machine learning techniques. By integrating various outlier detection models such as Isolation Forest, Local Outlier Factor (LOF), One-Class SVM, Random Forest Classifier, and a custom PAACDA algorithm, the system ensures high accuracy and adaptability across different data distributions. The architecture, designed using Django, allows seamless interaction between the user interface, application logic, and data processing layers. The user-friendly interface enables administrators to upload datasets, run multiple anomaly detection algorithms, and visualize the results with interactive graphs and metrics, all embedded within the web interface. The modular design of the system not only facilitates ease of maintenance but also promotes the addition of new models and functionalities. Overall, the system serves as a practical and efficient tool for anomaly detection in varied domains such as finance, healthcare, manufacturing, and cybersecurity.

REFERENCES

[1] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004, doi: 10.1007/s10462-004-4304-y.

[2] Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007, doi: 10.1016/j.comnet.2007.02.001.

[3] M. Jiang, P. Cui, and C. Faloutsos, "Suspicious behavior detection: Current trends and future directions," *IEEE Intell. Syst.*, vol. 31, no. 1, pp. 31–39, Jan. 2016, doi: 10.1109/mis.2016.5.

[4] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," 2014, *arXiv:1403.2877*.

[5] J. Gama, A. Ganguly, O. Omitaomu, R. Vatsavai, and M. Gaber, "Knowledge discovery from data streams," *Intell. Data Anal.*, vol. 13, no. 3, pp. 403–404, May 2009, doi: 10.3233/ida-2009-0372.

[6] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014, doi: 10.1109/tkde.2013.184.

[7] Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3634–3642, 2015, doi: 10.1016/j.eswa.2014.12.029.

[8] N. Jindal and B. Liu, "Review spam detection," in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 1189–1190.