Transformer based Anomaly Detection on Multivariate Time Series Subledger Data

Daksha Yadav
Amazon
dakyadav@amazon.com

Xiaoli Zhang
Amazon
zhasabri@amazon.com

Boyang Tom Jin
Amazon
boyanjin@amazon.com

ABSTRACT

Subledgers maintain detailed information about specific accounts or transactions in order to substantiate the general ledger. Subledgers provide a granular level of detail for financial reporting and analysis, which is especially essential for accounts receivables and payables. The size of subledgers can vary greatly depending on the complexity and volume of transactions and their size can also increase over time as more transactions are recorded. Depending on the size of a company, subledgers may record hundreds of billions of financially significant business events each year originating from its different legal entities. As many accounting customers rely on the subledger as the source of financial results, it is crucial to identify anomalies early on to minimize their impact and prevent further harm. To this end, we present a novel algorithm designed to analyze subledger transactions in a systematic manner. Our approach employs a modified transformer architecture for the task of anomaly detection in multivariate time-series data which relies on its ability to reconstruct input. Moreover, it utilizes the reconstruction loss as a priority value to emphasize data points that may be anomalous. We also enhance the anomaly score by incorporating seasonality and relationships between different attributes of the subledger. Experimental results demonstrate the efficacy of the proposed approach for different types of anomalies.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

anomaly detection, neural networks, accounting irregularities, subledger analysis, machine learning

ACM Reference Format:
Daksha Yadav, Xiaoli Zhang, and Boyang Tom Jin. 2023. Transformer based Anomaly Detection on Multivariate Time Series Subledger Data. In *In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 06–10, 2023, Long Beach, CA, USA.* . ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

SIGKDD, August 06–10, 2023, Long Beach, CA

ACM ISBN 978-1-4503-XXXX-X/23/08…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

A subledger, also known as a subsidiary ledger, is a type of accounting ledger that contains detailed information about specific types of transactions or accounts within a company's overall accounting system. It serves as a subset of the general ledger, which is the main accounting record that contains all financial transactions of a company. For example, a company might have a subledger specifically for accounts payable, which would contain detailed information about all payments made to vendors, such as invoice numbers, payment dates, and amounts. Another subledger might be dedicated to fixed assets, containing detailed information about all the assets a company owns, such as purchase date, cost, and depreciation.

Subledgers are used to maintain more detailed information about specific accounts or transactions than what is provided in the general ledger. They are often used to manage complex or high-volume transactions, and to provide a more granular level of detail for financial reporting and analysis. The size of subledgers can vary greatly depending on the complexity and volume of transactions they are used to manage. The size of a subledger also increases over time as more transactions are recorded. For example, a subledger for accounts payable may start small but grow rapidly as a company begins to work with more vendors and make more frequent payments. Depending on the size of an organization, it can record hundreds of billions of financially significant business events originating from its different businesses. These events represent different types of transactions, such as orders, shipment movements, invoice payments, and customer returns.

In many organizations, the subledger is the primary source for transaction details and is used to support the general ledger. Many accounting customers rely on it as the source of financial results. However, with a multitude of manually configured rules used in the subledger with ambiguous ownership and lack of coordination between various stakeholders, data issues frequently arise that require ticket escalations. These issues often stem from changes in column values (e.g., from "Shipping" to "shippingComplete") or missing column values. Even though the engineering team responsible for maintaining the subledger infrastructure is not responsible for the value of those fields or the configuration rules used to calculate those values, they still receive tickets and must investigate to confirm the correctness of their system.

In practice, anomalous journal entries can have a significant impact on any business team. These anomalies can occur due to a variety of reasons, including errors in source system code changes, incorrect filtering of transactions, fraudulent activities by buyers or sellers, and more. Moreover, delayed detection of these issues can lead to significant financial impact as these issues may remain

undetected for several days or even months, resulting in multi-million/billion dollar financial impacts.

In order to benefit both accounting customers and engineering teams, an automated anomaly detection or data pattern change detection system would be useful in notifying customers of any shifts in the underlying subledger traffic. Anomaly detection is an essential technique to help customers identify unusual patterns or behaviors that may indicate fraudulent activity, errors, or other issues that can negatively impact business performance, such as unsatisfactory customer experience or financial losses. However, because of the dynamic nature of the subledger traffic, maintaining a rule-based data monitoring system would be an operationally infeasible endeavor. Therefore, leveraging machine learning models would provide a sustainable and scalable solution that does not rely on fixed rules. The proposed model can notify accountants of changes in the subledger lines, thereby narrowing down the scope of the investigation conducted by the accountant at month-end. By detecting anomalies early, customers can take corrective action to minimize their impact and prevent further damage.

Over the years, a substantial amount of research has been conducted on detecting anomalies in time-series data. However, the subledger data is distinct from the traditional time-series datasets studied. Firstly, the number of attributes or dimensions in the subledger data is not commonly seen in the literature. Additionally, the large size of the data poses another challenge, with millions of rows being added on a daily basis. Furthermore, the multivariate time-series anomaly detection techniques described in the literature do not explicitly model the inherent relationships between various attributes or columns.

In this paper, we present a new algorithm that is specifically designed to analyze subledger transactions in a systematic manner. Its primary purpose is to identify any variances in the data over time and to provide customers with the necessary tools to conduct comprehensive investigations into these anomalies. To achieve this, we utilize a modified transformer architecture, which has been optimized for detecting anomalies in multivariate time-series data by leveraging its ability to reconstruct input data. Additionally, we use the reconstruction error to assign priority to data points that show high deviation and thus, may be anomalous. This enables the attention network to extract relevant features that aid in generating accurate reconstructions and we perform this in a two-stage training architecture to improve the algorithm's performance. Furthermore, we enhance the accuracy of the anomaly score by taking into account the seasonality and interdependence between different attributes in the subledger.

The paper is structured as follows: Section 2 describes the subledger data source in detail. Section 3 provides an overview of related research in the field of anomaly detection in time-series data. In Section 4, we explain our proposed methodology, including a modified transformer architecture and the use of reconstruction loss as a priority value. Section 5 details the synthetic data generation process used in this study. Finally, in Section 6, we present the results of our experiments, demonstrating the effectiveness of our approach.

2 DATA SOURCE

For this paper, we obtained access to a large private subledger dataset that contains billions of journal entries and is the authoritative source for transaction details to substantiate the general ledger of a large company. The subledger consists of more than one thousand attributes and increases in size by hundreds of millions of rows per day. The attributes contain varied information about the nature of the transaction, such as which source system posted it, what is the functional currency of the transaction, what kind of business activity it is (refund, sale, etc.) and so on. Different attributes may be critical for different subsidiaries and types of transactions. The journal entries are available by querying time-bounded partitions in a data warehouse. The data used for this paper was extracted using a SQL query in the data warehouse. Figure 1 shows a sample of the data stored in the subledger.

Each accounting customer is responsible for managing and reviewing a specific subset of the subledger data. Generally, at the end of each month, the customers perform reconciliation and check for the financial correctness of the accounts or businesses they are responsible for. However, sometimes discrepancies may be observed, and they need to investigate which transactions might have caused the issue and identify its root cause. Unfortunately, combing through millions of subledger transactions can be an arduous task and may take significantly longer than the short, time-sensitive duration of the month-end process. Therefore, in this paper, we propose an anomaly detection approach that can be run frequently (rather than waiting until the end of the month) so that customers can take corrective actions quickly at the time of an anomaly to minimize the impact of any issues.

3 RELATED WORK

The existing literature on anomaly detection has focused on two types of time-series data: univariate and multivariate. Univariate methods are designed to analyze and detect anomalies in time-series data with a single data source, while multivariate methods consider multiple time-series together.

Classical techniques: These anomaly detection algorithms commonly utilize classical techniques such as k-Means clustering [2], Support Vector Machines [13], or regression models [4] to model the time-series distribution. Additionally, other methods use wavelet theory or signal transformation techniques like Hilbert transform, PCA, process regression, and hidden Markov chains are used to model time-series data [10]. The GraphAn technique [1] converts time-series inputs into graphs and uses graph distance metrics to identify outliers. The isolation forest technique, on the other hand, uses an ensemble of isolation trees [8] to partition the feature space recursively for outlier detection. Finally, classical methods use variants of Auto-Regressive Integrated Moving Average [16] to model and detect anomalous behavior. However, auto-regression-based approaches are not frequently used for anomaly detection in high-order multivariate time series due to their inability to capture volatile time-series efficiently.

Deep Learning based techniques: Most state-of-the-art anomaly detection techniques rely on deep neural networks. With respect to analyzing subledger and detecting anomalies, autoencoders have been trained and the trained network's reconstruction error for a

Date	Subsidiary	Country	Activity	Component	Currency	Amount	Invoice Type	Quantity	...
2023-03-15	AAAAA Inc.	US	Sale	Tax	USD	12.49	-	3	...
2023-03-16	BBBBB Inc.	JP	Refund	Invoice	JPY	99240	Partial	100	...
...	...	...	...	...	...	...	...	...	...

Figure 1: Two sample journal lines in the subledger. The keys and values shown in this image are synthetically generated for visualization purposes but reflect the actual type of data and format that is stored in the private subledger dataset.

The above figure shows two journal entries from 2 different dates, subsidiaries, country codes, type of business activity, business component, transaction amount and currency, etc.

journal entry along with the individual attribute probabilities is utilized [11]. Building on this, [12] used adversarial autoencoder networks to learn semantic representation of real-world journal entries and use that to detect accounting anomalies. In other works, the LSTM-NDT [5] method uses an LSTM-based deep neural network model that forecasts data for the next timestamp based on the input sequence used as training data. However, being a recurrent model, such models can be slow to train for long input sequences. The MAD-GAN [6] uses an LSTM-based GAN model to model the time-series distribution using generators. This work uses not only the prediction error but also the discriminator loss in the anomaly scores. The CAE-M [17] uses a convolutional autoencoding memory network that passes the time-series through a CNN, and the output is processed by bidirectional LSTMs to capture long-term temporal trends. Recently, TranAD [14] has leveraged transformers to learn the inherent data distribution of regular data and uses reconstruction errors as an indicator of anomalies. However, TranAD does not effectively encode the relationships between different dimensions of the data.

4 PROPOSED APPROACH

4.1 Problem Definition

Our work deals with a multivariate time-series, which is a sequence of timestamped observations or data points of size T, denoted as $\tau = \{x_1, x_2, ..., x_T\}$. Each of the data points, x_t, is recorded at a particular timestamp t where $x_t \in \mathbb{R}^d$ and the number of attributes or features $(d) > 1$ for the multivariate setting.

The problem of anomaly detection is defined as follows. Given a training input time-series τ and an unseen test time-series τ_{test}, which has the same modality as the training series, our objective is to predict $y = \{y_1, y_2, ..., y_t\}$ where we use $y_t \in \{0, 1\}$ to indicate whether the data point at the t-th timestamp in τ_{test} is anomalous, 1 representing an anomaly.

4.2 Model details

Transformers are well-established deep learning models that have been successfully applied to a wide range of natural language and vision processing tasks [7]. In our work, inspired by TranAD [14], we employ a modified transformer architecture for the task of anomaly detection in time-series data. Similar to other encoder-decoder models, the transformer takes an input sequence and applies multiple attention-based transformations. Our model is composed of two transformer encoders [15] and two decoders, producing two outputs Out_1 and Out_2. The proposed approach is based on the hypothesis that if we train an encoder-decoder network to accurately encode and then reconstruct *normal* data, then we can use the increase in reconstruction error as an indicator of a possible anomaly. This can guide the attention network to emphasize these sub-sequences with high deviations.

To capture the dependence of a data point x_t at timestamp t, we use a window of length L, defined as $W_t = \{x_{t-L+1}, ..., x_t\}$. Rather than using the entire τ as the training input, we use W for model training and W_{test} as the test series. Additionally, we consider the time slice of a series τ up to the current timestamp t and denote it as $Curr_t$. For experimental evaluation, L is set to 7.

Using the Transformer model, we aim to reconstruct each input time-series window by leveraging its encoder-decoder network at each timestamp. However, standard encoder-decoder models may struggle to capture short-term trends and may miss anomalies if the deviations are too small. To address this issue, we predict the reconstructed window in two stages which are described subsequently.

During the first stage, the primary objective of the first decoder is to create an estimated reconstruction (Out_1) of the input window W. This reconstruction provides the basis for the priority value $p = ||Out_1 - W||$. The dimensionality of p is same as W_t. The Transformer encoder utilizes the priority value to assist in identifying temporal patterns by focusing on the sub-sequences that exhibit high deviations. This process enables the attention network to extract relevant features that can aid in generating accurate reconstructions. As a result, the output of the second stage is conditioned on the deviations generated from the first stage, which helps to further refine the accuracy of the reconstructed sequence.

During the second phase, we leverage p obtained from the first decoder. This value is used to create a priority matrix for the second phase. By using this priority matrix, we rerun the inference process to generate the output of the second decoder, $Out_{2-updated}$. The priority value generated in the first stage serves as a prior that modifies the attention weights in the second stage, giving higher activation to specific input sub-sequences and enabling the extraction of short-term temporal trends. This two-stage auto-regressive inference style provides multiple benefits. Firstly, it amplifies deviations by using reconstruction error as an activation in the attention part of the encoder shown to generate an anomaly score. Secondly, it prevents false positives by capturing short-term temporal trends in the Window encoder. The second decoder tries to distinguish between the window W_t and the reconstruction loss.

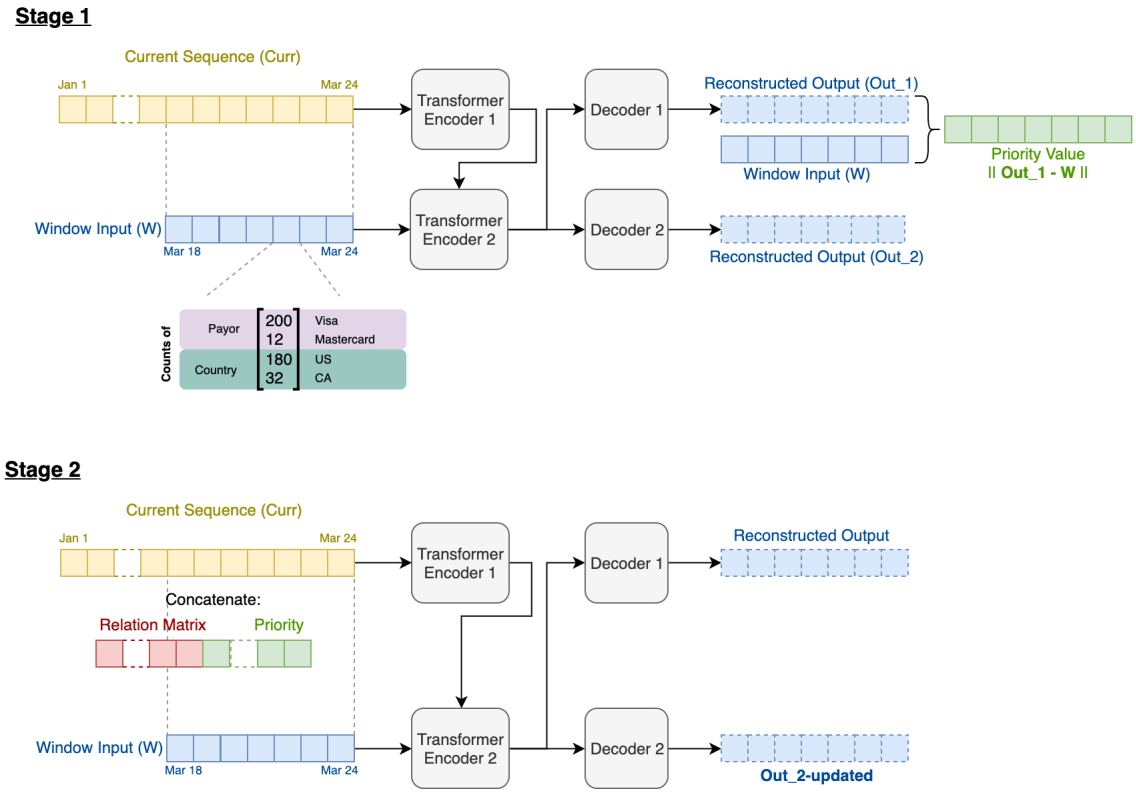

Figure 2: An overview of the proposed methodology for transformer based anomaly detection on subledger data.

During the first stage, the objective of the first decoder is to create an estimated reconstruction (Out_1) of the input window W. This reconstruction provides the basis for the priority value $p = ||Out_1 - W||$. During the second phase, we leverage p obtained from the first decoder. This value is used to create a priority matrix for the second phase. By using this priority matrix, we rerun the inference process to generate the output of the second decoder, $Out_{2-updated}$. Apart from relying on the Transformer encoder to learn the inherent relationships between the d dimensions of the timeseries, we explicitly use that information as a prior in Stage 2. We introduce Relation Matrix (R) which is computed by learning joint-probability distribution of top-k most frequently occurring values for each pair of attributes. R and p are converted to a one dimension and concatenated with the input.

4.3 Encoding Inherent Relationships between Subledger Attributes

Apart from relying on the Transformer encoder to learn the inherent relationships between the d dimensions of the timeseries, we explicitly use that information as a prior in Stage 2. We introduce a $k \times k$ size Relation Matrix (R) which is computed by learning joint-probability distribution of top-k most frequently occurring values for each pair of attributes. The logic behind that is in most cases, these attributes/columns are not independent variables. For instance, the attributes *Country = UK* along with the attribute *Subsidiary = AAAAA London Inc.* are more probable to occur together and an anomaly in one of those attributes should be a good indicator of a possible anomaly in the other attribute. Specifically, if there was a missing remittance in AAAAA London Inc., we would expect to see cascading effects leading to multiple co-occurring anomalies in multiple accounts in the UK.

4.4 Adaptive Anomaly Score Computation

In this sub-section, we describe the process of anomaly score computation and anomaly label assignment. For unseen test data W_{test}, the anomaly score a is computed by adaptively weighing the reconstruction output from the two decoders:

$$a = w_1 * ||Out_1 - W_{test}||_2 + w_2 * ||Out_{2-updated} - W_{test}||_2 \quad (1)$$

The values of weights w_1 and w_2 are computed empirically for different datasets depending on the importance to be placed on the two decoders.

Next, the anomaly scores are adjusted based on seasonality in order to reduce false positives caused by predictable variations. Intuitively, if the data point at a given timestamp has been labeled anomalous by the above approach and a similar trend has been observed seasonally (previous hour, week, or month), then the anomaly score for the given timestamp is readjusted by multiplying by the adjustment factor β (<1) to reduce the anomaly score.

After computing the adjusted anomaly score for each of the d dimensions, the next step is to assign the combined/joint anomaly score for a given timestamp, which is done by averaging the anomaly scores across d dimensions. If the combined anomaly score is greater than a pre-determined threshold, then we label that timestamp as 1 (or anomalous).

5 SYNTHETIC DATA GENERATION

To ensure the accuracy of evaluation, one of the biggest challenges in anomaly detection is the scarcity of labeled data, which is crucial to verify the robustness and flexibility of proposed methods. While human feedback can serve as a ground-truth, it is manual, subjective, time-consuming, and limited, and thus, may not encompass the full range of scenarios that the model may encounter in the real-world, rendering it inadequate for comprehensive evaluation.

In such circumstances, synthetic data generation and subsequent anomaly generation can offer a solution by producing a more diverse and extensive set of anomalous data with tailored levels of complexity and variability. This approach reduces the reliance on human feedback to provide ground-truth data. Therefore, in this paper, we generate synthetic subledger data and use it to generate a wide range of synthetic anomalies for evaluation purposes. The methodology is explained in detail below:

5.1 Synthetic Subledger Data Generation

To accomplish this, we begin by sampling actual subledger data for a particular company code and account number combination. This reflects the real-world situation where an accounting customer employs these filters to examine relevant reports. We sample data from Jan 1, 2022 to Feb 15, 2023 with sum of quantity of items being the primary value of interest. For each transaction, certain subledger attributes such as the source system, business namespace, and financial component were selected based on their relevance in encoding critical information about different financial events.

In order to remove any apparent peaks/anomalies in the data, quantiles are computed and the data between 1% to 98% quantiles are selected. Next, this sampled data is preprocessed. Most of the attributes are categorical in nature and therefore, are converted to one-hot encoding. After this, column-wise normalization of the data is also performed. As the next step, this data is now used for training CTGAN (Conditional Tabular Generative Adversarial Network) [3] to learn the inherent data distributions. CTGAN is trained for 500 iterations and the trained model is then used to generate data for 365 days. For each day, 100 different timeseries data are created to ascertain the efficacy of the proposed approach with statistical significance.

5.2 Synthetic Anomaly Generation

Given synthetic subledger data generated from the previous step, anomalies are synthetically injected to the data. Based on review of previous real anomalies, synthetic anomalies are generated based on the following variables:

- Anomaly fraction (f): This indicates the percentage of anomalies in the data.

$$f = 100 * \frac{\text{Number of anomalous data points}}{\text{Total number of data points}} \tag{2}$$

  Intuitively, smaller value of f denotes rare occurrence of anomalies as compared to larger value of f indicating more-frequently occurring anomalies. We select three different values of f: 1%, 5%, and 10%.
- Anomaly scale (s): This indicates the scale/size of the anomaly as compared to the regular/normal data and is used as a multiplier. The range of the normal data is scaled by this factor to inject anomalies. A positive number > 1.0 is needed to create meaningful anomalies/outliers. We select three values of s: 1.5, 2, and 3 .

6 EXPERIMENTS AND RESULTS

The proposed approach is evaluated on the synthetic subledger data described above. Results on the private subledger dataset are not released for confidentiality. We also perform comparative analysis with existing anomaly detection algorithms including TranAD [14], LSTM [5], CAE_M [3], and MAD GAN [6]. We use F1-score to evaluate the detection performance of all the models. We utilize 80%-20% train-test split to train the models and obtain their performance. As mentioned above, for each experiment, we train on the 100 sets of 80% training data and report average results for statistical significance.

Another benchmark dataset, Multi-Source Distributed System (MSDS) [9], is used to evaluate the performance of the proposed dataset. It is a recent multi-source data composed of distributed traces, application logs, and metrics from a complicated distributed system . This dataset was specifically built for training machine learning models, including automated anomaly detection, root cause analysis, and mitigation. It contains 146,430 training datapoints along with 146,430 test data points with 5.37% anomalies present.

6.1 Results on Synthetic Subledger Dataset

6.1.1 Impact of Anomaly Fraction. We ran different experiments with varying levels of anomalies injected into the subledger data. We selected three different values of anomaly fraction (f) = 1%, 5%, and 10% and the results are shown in Table 1 and Figure 3. When the anomaly fraction is just 1% of the total data, we observe that the proposed approach produces 0.74 as the F1-score, outperforming the other approaches. Similar performance is observed when f = 5% and 10%.

6.1.2 Impact of Anomaly Scale. From Table 2 and Figure 4, it is observed that at varying anomaly scales, the proposed algorithm outperforms the existing approaches. When the anomaly scale is 1.5 times, the proposed algorithm yields 0.74 as the F-1 score, outperforming the other algorithms. It is noted that when the anomaly scale increases, i.e. the anomaly becomes more apparent, the algorithm performs better. Intuitively, the higher the anomaly scale, the easier it is to detect them. Therefore, when the anomaly scale is 2, the proposed approach produces 0.8 as the F1-score as compared to when the anomaly scale is 3, where it produces 0.83 as the F1-score.

6.2 Results on MSDS Dataset

Table 3 shows the performance of the proposed approach and other state-of-the-art techniques on the MSDS dataset [9]. This dataset

Table 1: Results (F1-score) of the proposed approach for multivariate anomaly detection with different anomaly fractions on the Synthetic Subledger Dataset.

Algorithm	Anomaly Fraction=1%	Anomaly Fraction=5%	Anomaly Fraction=10%
TranAD [14]	0.72	0.70	0.73
LSTM [5]	0.65	0.69	0.70
CAE_M [3]	0.66	0.66	0.64
MAD GAN [6]	0.71	0.65	0.72
Proposed	**0.74**	**0.73**	**0.75**

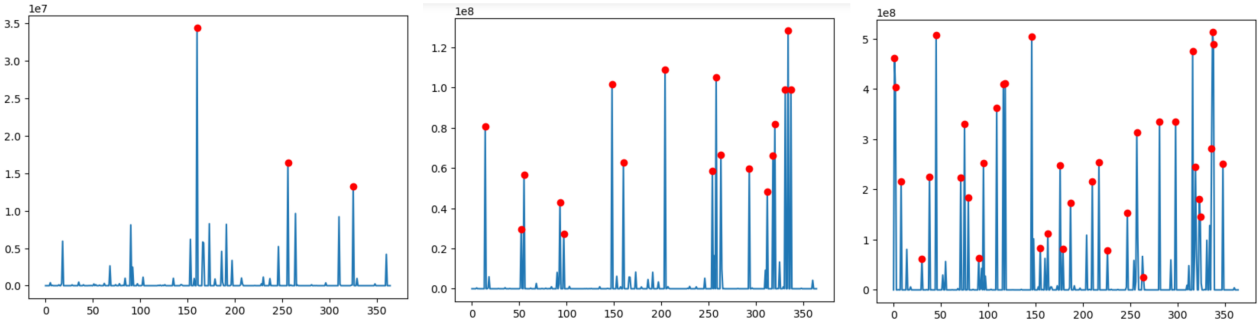

Figure 3: Demonstrating the performance of the proposed approach for multivariate anomaly detection on Synthetic Subledger Dataset with different fractions of anomalies. The x-axis represents the day range (starting from 0 to 364). The y-axis represents the value of the sum of quantity on a daily basis. The red dots denote the anomaly prediction by the proposed approach.

Table 2: Results (F1-score) of the proposed approach for multivariate anomaly detection on Synthetic Subledger Dataset with different anomaly scales.

Algorithm	Anomaly Scale = 1.5	Anomaly Scale = 2	Anomaly Scale = 3
TranAD [14]	0.72	0.77	0.80
LSTM [5]	0.67	0.69	0.72
CAE_M [3]	0.65	0.70	0.71
MAD GAN [6]	0.60	0.63	0.66
Proposed	**0.74**	**0.80**	**0.83**

Table 3: Results (F1-score) of the proposed approach on MSDS [9] benchmark dataset.

Algorithm	F1-Score
TranAD [14]	0.93
LSTM [5]	0.89
CAE_M [3]	0.91
MAD GAN [6]	0.76
Proposed	**0.95**

contains data from ten different dimensions/attributes. It is observed that the proposed approach outperforms other state-of-the-art algorithms for anomaly detection. The proposed approach produces the highest F1-score of 0.95, outperforming TranAD, which yields 0.93 as the F1-score. For this dataset, the optimal value of w_1 and w_2 are 0.4 and 0.6, respectively which are computed based on experimental evaluation.

CONCLUSION

Subledgers play a crucial role in maintaining detailed information about specific accounts or transactions, providing a necessary level of granularity for financial reporting and analysis. Given the reliance of accounting customers on the subledger as a source of financial results, it becomes imperative to identify anomalies at an early stage to minimize their impact and prevent further harm. In practice, anomalous journal entries can have a significant impact on any business team. These anomalies can occur due to a variety of reasons, including errors in source system code changes, incorrect

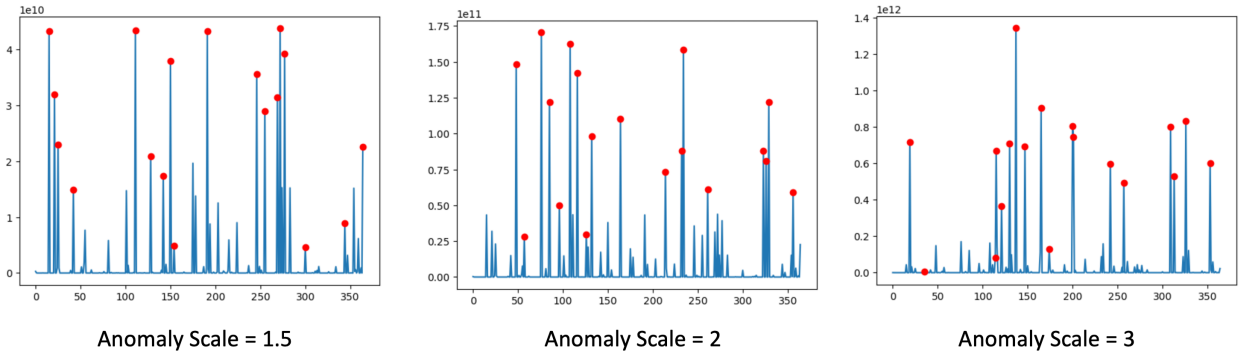

Figure 4: Demonstrating the performance of the proposed approach for multivariate anomaly detection on Synthetic Subledger Dataset with different scales of anomalies. The red dots denote the anomaly prediction by the proposed approach.

filtering of transactions, fraudulent activities by buyers or sellers, and more. In this research, we have presented a novel algorithm specifically designed to analyze subledger transactions in a systematic manner. Our work enables proactive identification of granular anomalies, allowing accounting customers to take corrective measures promptly and minimize their impact, ultimately preventing further financial impact. Our approach utilizes a modified transformer architecture, which has shown effectiveness in anomaly detection tasks in multivariate time-series data by leveraging its ability to reconstruct input. By utilizing the reconstruction loss as a priority value, our algorithm emphasizes data points that may indicate anomalies. Additionally, we have incorporated seasonality and relationships between different attributes of the subledger to enhance the anomaly score. Experimental results have demonstrated the efficacy of our proposed approach across various types of anomalies. In conclusion, our research offers a valuable contribution to the field of subledger analysis and anomaly detection, with significant implications for improving financial data integrity and decision-making processes in organizations relying on subledgers for their accounting operations.

REFERENCES

[1] Paul Boniol, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2020. Graphan: Graph-based subsequence anomaly detection. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2941–2944.

[2] Sanjay Chawla and Aristides Gionis. 2013. k-means–: A unified approach to clustering and outlier detection. In *SIAM international conference on data mining*. SIAM, 189–197.

[3] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. 2018. Conditional generative adversarial network for structured domain adaptation. In *IEEE conference on computer vision and pattern recognition*. 1335–1344.

[4] Weiming Hu, Jun Gao, Bing Li, Ou Wu, Junping Du, and Stephen Maybank. 2018. Anomaly detection using local kernel density estimation and context-based regression. *IEEE Transactions on Knowledge and Data Engineering* 32, 2 (2018), 218–233.

[5] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *ACM SIGKDD international conference on knowledge discovery & data mining*. 387–395.

[6] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *International Conference on Artificial Neural Networks*. Springer, 703–716.

[7] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open* (2022).

[8] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *IEEE international conference on data mining*. IEEE, 413–422.

[9] Sasho Nedelkoski, Jasmin Bogatinovski, Ajay Kumar Mandapati, Soeren Becker, Jorge Cardoso, and Odej Kao. 2020. Multi-source distributed system data for AI-powered analytics. In *Service-Oriented and Cloud Computing: 8th IFIP WG 2.14 European Conference, ESOCC 2020*. Springer, 161–176.

[10] Animesh Patcha and Jung-Min Park. 2007. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks* 51, 12 (2007), 3448–3470.

[11] Marco Schreyer, Timur Sattarov, Damian Borth, Andreas Dengel, and Bernd Reimer. 2017. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv preprint arXiv:1709.05254* (2017).

[12] Marco Schreyer, Timur Sattarov, Christian Schulze, Bernd Reimer, and Damian Borth. 2019. Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks. *arXiv preprint arXiv:1908.00734* (2019).

[13] Vasilis A Sotiris, W Tse Peter, and Michael G Pecht. 2010. Anomaly detection through a bayesian support vector machine. *IEEE Transactions on Reliability* 59, 2 (2010), 277–286.

[14] Shreshth Tuli, Giuliano Casale, and Nicholas R Jennings. 2022. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284* (2022).

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[16] Asrul H Yaacob, Ian KT Tan, Su Fong Chien, and Hon Khi Tan. 2010. Arima based network anomaly detection. In *International Conference on Communication Software and Networks*. IEEE, 205–209.

[17] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. 2021. Unsupervised deep anomaly detection for multi-sensor time-series signals. *IEEE Transactions on Knowledge and Data Engineering* (2021).