International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V5I3P125
Eureka Vision Publication | Volume 5, Issue 3, 241-249, September – 2024

Original Article

Graph Neural Networks (GNNs) and Parallel Autoencoders for Heterogeneous Enterprise Log Mapping

Mohammed Saad Javeed
Network Engineer, Oakbrook Terrace, IL, USA.

Abstract - These numerous kinds of logs cover all the various applications, infrastructure, security and enterprise Information Systems. But more and more often you will get nights plagued with lots of cross domain, chaotic data structures and clunky log format for situational awareness, anomaly detection and compliance based auditing. Considering the need of intelligent mapping, reconstructing and analysing the various enterprise logs, this research proposes Integrated Graph Neural Networks (GNNs) and Parallel Autoencoders (PAEs) for Mapping Heterogeneous Enterprise Logs. In this paper, the concept is to leverage the parallel autoencoders to model the structure and the relationships between context elements from the different sources and fuse with the graph representation learning approach to obtain the logs. The learnt log undergoes the unsupervised reconstruction loss and this approach is capable of capturing the variant behaviour of log patterns, and doesn't need a large amount of labelled data to cover all these possibilities. Moreover, the inter dependencies between systems that cross multiple platforms/cross domain networks can be identified using GNN based mapping. The analytics base can be used to build logs from the networks and store them in silos, and can detect abnormal behaviors, with the risk assessment for compliance purposes. The information of distributed features reconstruct is to be merged with the information in relational graph learning to be fed in the samplable conceptual structure to improve enterprise log visibility, security intelligent monitoring & audits. The vineyard highlights how the heterogeneous systems can share a common enterprise management of logs, which can be achieved by leveraging and synergies brought by unsupervised learning and graph-based analytics.

Keywords - Graph Neural Networks (GNNs), Parallel Autoencoders, Unsupervised Reconstruction Loss, Heterogeneous Enterprise Log Analysis, Compliance-Risk Auditing, Siloed Network Log Parsing, Cross-Domain Network Topologies.

1. Introduction

In today's enterprise information systems the volume of log data from various distributed systems, operating systems, databases, cloud and network devices and security infrastructure is huge. These logs carry valuable data regarding systems' activity and dependency, security violation and performance of applications. In enterprises, however, many are using a mix of technologies and self-governing infrastructure pieces that produce a variety of log formats, lack common events or representation of those same events and use a collection of monitoring approaches. This renders it difficult for any of the parties to develop a stable and stable definition of "logs" and of any sort of relationship between the events from different organisations.

Bidirectional Encoder Representations from Transformers (BERT) [1] was proposed by Devlin et al. which applies a bidirectional contextual learning to get thunks representations, dense text representations. This kind of contextual representation gainful substrates to comprehend enterprise log messages with a diversity of heterogeneous channels. It is an all-round framework for log mapping that should, therefore, look to extract contextual features, along with creating a relation model between structures.

Anomaly detection, using deep learning has also shown great promise in detecting abnormal system activities in sequential log-analysis. In order to detect deviations from previously learned behaviour in a model of system log sequences [2] [3].

The larger ones of them are the transformation of the log data to structured data (automated analysis of logs). In the typical enterprise application, your logs might be semi-structured, unstructured, may contain several points of timestamping, event-id's and may contain multiple parameters (which changed), system information and other descriptions of actions the app took and/or failed to take. He et al. [4] suggested a tree structured online log parsing (Drain) approach with a fixed-depth. Using this approach they are able to efficiently extract the log templates from the log messages dynamically-created during the runtime of the system, to assist in converting unstructured information to a structured representation of the events. These parses can serve as excellent start points when it comes to enterprise log processing toward a non-homogeneous parsing.

A second crucial side in relation to the inferences of time-states in enterprise logs streams is sequential studying. The biggest benefit of LSTM is that they can prevent the learning of long-term dependencies when learning a sequential information such as the problem. It can easily be adapted as a good base for modelling temporal aspects of the system activities

as the knowledge about the events can be extracted in a way that lie ahead of the activities in the system. Modelling however will involve temporal modelling and contextual representation and in such cases mechanisms can be used to recognise that there is a structural relationship between the enterprise parts. This encourages parallel usage of Autoencoder streams and Autoencoders in an analysis framework, like GNNs.

The general goal of such work is the development of a complete analytical framework which makes it possible to complement such a log, stemming from different sources with a compliance oriented risk-analysis, to identify anomaly including unsupervised learning processes. Our proposed method focuses on four (all inter-related) capabilities: heterozygous log parsing, parallel features reconstruction (based on autoencoder) graph-based dependency mapping and analytical interpretation, concentrating on compliance risks. All of these abilities represent the possibility of solving enterprise monitoring challenges in a conceptual way: a monitoring environment consists of many different components and pushing the limit of the “intelligent” log management in a distributed networking environment is essential.

2. Related Work

In the field of log analysis for heterogeneous enterprise, the traditional log parsing & stat based log analysis is migrated to deep learning-based log analysis – log representation, sequence modelling and anomaly detection without the need for labelled data. Collection of these research is ongoing and it is very crucial in order to understand the unstructured messages in log, find anomaly event sequence and obtain contextual information among massive and complex system log. Yet, for modelling relationships between dissimilar enterprise modelling contexts the mixture of parallel (representation) learning and graph based modelling is an important one to consider.

In 2018, Devlin et al. [1] have been introducing it an approach which represents language bidirectionally, Bidirectional Encoder Representations from Transformers (BERT), capturing the relationship between words in a sentence with it. This was a model which stressed the significance of embedding the text information in accordance with the context and this has been followed as a model in subsequent studies on text-information understanding. Everything that deals with technical communications, imperative event description and event identifiers and operational expressions on which from which its meaning can be inferred. In this regard representations of language in some "contextual" sense makes sense and is thus useful in converting to meaningful feature vectors. However, such a generic purpose LDM is not able to say anything about the relationship of the parts of the distributed infrastructure or describe any interactions between the log-emitters of different types.

Du et al. [2] have introduced a deep-learning approach called DeepLog which features anomaly detection and diagnosis from system logs at the system level. For this they employ a technique known as “Long Short-Term Memory” (LSTM) networks, which learn patterns in sequential events of the logs, and search for the sequential events on the bases of what they learned. Can be useful in an enterprise scenario where events occurring on top of various applications, servers and networks are related via some sequential processes. Yet, most of the focus of sequence oriented analysis is on the order of events; it lacks a good mechanism for expressing complex associations amongst different kinds of Logs, infrastructure and distributed components when they are involved in describing the activity. For this reason, we believe that unifying temporal learning with a relational graph resulted in the idea of this work.

To achieve this new attempt Guo et al. [3] proposed LogBERT, a BERT-based Log Anomaly Detector that performs Log anomaly detection. To recognize the abnormal pattern from the log sequences, we propose Developed, as a double aspect work, which are involved both contextual representation and concept in the self-supervised learning. One of the great insights from LogBERT is that although it do not necessarily need a full-fledged collection of manually annotated training data, transformer based models have a potential of making a difference in anomaly detection domain by offering an interpretation of the relationship between, and among, the anomaly events in the log. In particular, considering log heterogonous enterprises, these few “anomalies” or “irregularities” could involve log production throughout the applications. If all this information (context orientation) is stored in a sequential way then it is in topology/dependency of enterprise resources. Can be used in conjunction with graph-based learning, to obtain a broader understanding of the way that events, systems and operational entities are related.

He et al. developed an online log parsing algorithm called Drain in [4] that is based on a fixed depth tree structure. All the logs messages are collected and then the templates are created with the clustered dynamically created Log messages.

To overcome the learning long-range dependency problem of an RNN, however, Hochreiter and Schmidhuber [5] have proposed LSTMs (long short-term memory networks). They consist of memory structures, as well as control structures to regulate the flow of information between the logical inputs. Thus, LSTM network exercise an important role in Log Analysis as there may be some temporal relations and operation repeating behaviours among the life cycle of the logs. They can contain relevant historical data which can be used to analyse changes in the system activities and consecutive deviations. However, modelling with the conventional LSTM can capture most of the relations between the time series; there is no explicit graph

structure to model the relations between systems. This sort of separation is essential while enterprise logs are generated from various related programmes, databases, community gadgets and security platforms.

Huang et al., [6] first introduced hierarchical transformer-based nesting to improve the anomaly detection in system logs, which they called HitAnomaly. Uses a hierarchical modelling method which considers a set of log information on several levels in order to consider complex event patterns for analysis. The use of transformer based architecture are advantageous in modelling the contextual dependency, particularly when the different representations of event in the sequences are varied. There is a gap however, with respect to the requirements of the analysis for the hierarchical Sequence Analysis and Cross-Domain Relationship Mapping techniques. An example of such an approach that a graph based component could augment would be explicit representation of application to database to user to network segment dependencies.

Landauer et al. [7] reviewed in detail the log anomaly detection techniques using deep learning approach. They gave an overview on a number of learning architectures, preprocessing, as well as anomaly identification approaches and research characteristics. The survey provides an overview of deep learning approaches to detect abnormal system behaviour, such as transformer-based, autoencoders and recurrent approaches, and an increasing participation in recent circumstances. It further emphasizes the importance of focusing on logistics of the preprocessing and/or representation options in log analysis solutions.

Le and Zhang [8] performed a logbased anomaly detection that did not involve log parsing. The problem with the existing detection methods that their approach tackles is that they are based on well-known parsing rules, and they seek to explore directly the analysis of data in log messages. This is key as in the heterogenous enterprise environment, often the logs are dynamically changing, as are the key message constructs—with respect to how they are defined. Flexible parsing (which has to be done to allow the inclusion of variables on meaning and relation between words/sentences from different entries) could possibly be reduced, but there has to be some.

LogAnomaly proposed by Meng et al. [9] is an unsupervised approach to sum up the anomalies belonging to unstructured log sequential and quantitative. In this study not only the prediction of the sequence is discussed but also the deflections from the usual sequence and significant quantitative deflections. For an enterprise application both kinds of analytical view have valuable information because the unnormalized activities would not only constitute an abnormal number of events, but also an abnormal pattern of events. However, local levels of analysis of events and activities might not agree with other levels of more abstract and distant events and relationships between activities that are more diffused and distributed.

Mikolov et al. [10] came up with efficient methods to get the representations of words in vector space. They introduced pioneering techniques that learn word embeddings that can be used in distributed manner and in such technique, words with similar functions in context could have meaningful vectors. These representations, obtained as a result, can then be used to map the log from the textual to the numerical representation that can be used as input to the machine-learning algorithms. But, in case of more complex meanings of events and “enterprise infrastructure” relations between enterprise-infrastructure components, other modes of representation might need to be used in a word level representation.

But very importantly Nedelkoski et al. [11] applied classification based approach while analyzing the anomaly detection for logs, where they considered their self-attention. Everything they work on, such as attention mechanisms, can help them determine what information is relevant or pertinent to the log message, and help them classify the abnormal behavior. Self-Attention has helped to prove this point and to show some key components of a text, without the traditional sequential processing. However, frequently they are not communicated at all, or inadequately communicated and if a classification-based approach is applied they are sparse in complex enterprise applications.

To analyse the information in large scale computing system logs Oliner and Stearley [12] studied 5 system logs. They have really shown the importance of keeping logs to gain a better understanding of the behaviour of the system, when it went wrong and analysing infrastructure reliability. Also shows variation and complexity of log information with real world contexts. This 'base-line' view evokes a need for analytical approaches that will accommodate large volumes of operational logs containing a diverse range of logs, not merely a single type of log.

Global Vectors for Word Representation glove is a world embedding approach, which is based on co-occurrence statistics of words across all the world and vector approaches. Word representations can be defined by statistical information in this corpus, thus capturing some word relations in semantics. These types of embeddings can play a role in converting technical messages tokens to numerical features for Enterprise Log Analysis. Static word embeddings, however, are not enough to represent time-dependent dependencies of events and graph-like relationships between distributed systems.

Zhang et al. [14] suggested log data-based anomaly detection called LogRobust to deal with anomalies with unstable log conditions. Their research considers the challenges of log format diversity and the changing message structure, which makes it more difficult to be identified by traditional log-based approach. Knowing how to cope with these changes can be pivotal in

enterprise networks, where software changes, configurations, and other changes in the network can all influence log generation; or where infrastructure changes can also influence log production due to frequent software changes. In their contribution, they stress the need for the development of flexible data representations for evolving operational data.

3. Parallel Representation Learning and Graph-Based Heterogeneous Log Intelligence Framework

The proposed scheme ParAE-GNN Architecture for Cross-Domain enterprise Log Intelligence comprises of features learning from source, anomaly detection using reconstruction and enterprise relation modelling with the graph. It fully supports the logging of the heterogenous enterprise logs generated by distributed applications, databases, OS, network devices and security products in a uniform structure for analysis purposes. Unlike seeing each log source as a standalone data pipe the framework takes into account the attributes of the different log categories as well as the relationships between the events that span enterprise infrastructure. But it consists of 6 steps which are dependent on each other: defect items recognition based on reconstruction from abnormal vectors in Gaussian process, compulsory construction of graph based on abnormal items in third step, graph relationship analysis based on GNN in fourth and fifth steps, respectively, and interpretation of defect items based on compliance risk in sixth step. Figure 1 illustrates the overall architecture of the proposed framework, connecting heterogeneous log acquisition, parallel representation learning, graph-based analysis, and compliance-risk interpretation

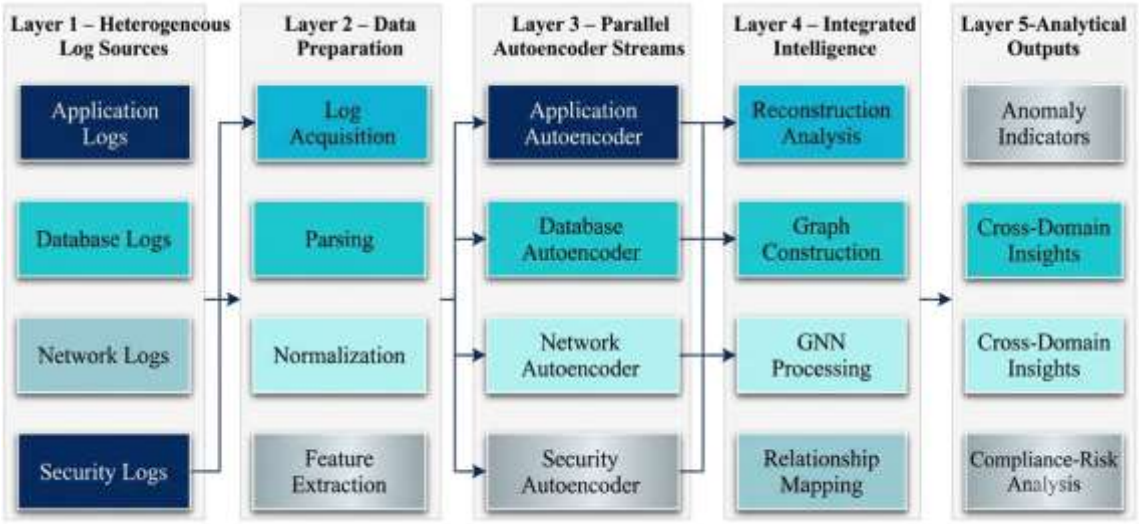


Figure 1. Integrated Parallel Autoencoder–GNN Framework for Heterogeneous Enterprise Log Mapping

3.1. Heterogeneous Log Acquisition and Normalization

In the first phase the data is aggregated at the different data sources in the enterprise, these can be applications or application servers, DBMSs, network devices, cloud, authentication systems, security monitoring systems etc. These sources produce a variety of structures and formats of logs; time, event type and depth of description. All these oscillations give difficulties in rendering to a stable analytical form.

The metadata and infrastructure references are comprised of information about the source identifier of the event, categories and level of seriousness. In this step the logs are translated into structured representations that involves no loss of potentially relevant information that could be useful for later analysis as context. Normalized records are then passed to appropriate parallel learning streams, based on the constructions and the kind of their sources.

3.2. Parallel Autoencoder-Based Feature Representation

Next, the parallel stream of autoencoders learn a compression codes of each stream of categories during the 2nd stage. Events at the application level or network-level events, database-level events, or security level events may have unique attributes that might not be best suited to a single learning architecture. The trick is, we will need to store an autoencoder for every type of log/source of group to make this idea practical as a system.

Autoencoders have two components – The Encoder and the Decoder. The normalized log features are encoded in the encoder space to a more dimensional latent space and the decoding space tries to re-construct the original one from the lower dimensional space. The encoder takes in its log stream, and determines recurring patterns of structures and/or behaviours. Reconstructed learned representation is then given to the decoder to determine whether there is any difference in the pattern observed during training or not. The source-specific learning arrangement is depicted in Figure 2, highlighting parallel autoencoder streams and their contribution to unified feature representation.

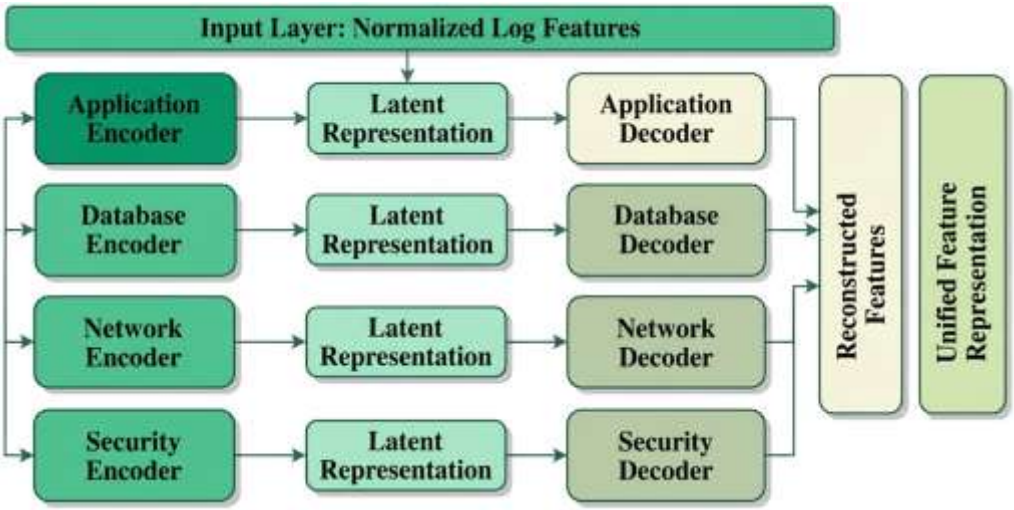


Figure 2. Source-Specific Parallel Autoencoder Architecture

These parallel streams can allow one stream to learn the representation of the property of one source WHILE another stream is learning the representation of the property of another source, without negatively affecting a representation-learning approach that is consistent across all the streams. For instance, an application-log branch could contain data regarding the usual request operations, aimpoints, application errors, a network-log branch could contain data regarding the behaviour of the connections, occurrences of traffic and communication patterns. Similarly, the database and security branches can obtain patterns to learn – for database, for transaction activities and for authentication events. The latent representations thus obtained are then used for embedding of a low dimensional representation that will also be useful for further anomaly analysis and that will be used for constructing a graph in a space.

3.3. Reconstruction-Based Anomaly Identification

For stage 3, it is possible that there is a relationship between the reconstruction loss and potentially abnormal log patterns. Autoencoders build up the ability to recreate the typical input features. However, the number of log record to be recorded in the system is very small, if the log record recorded in the system is dissimilar to the pattern learnt by a trained Autoencoder, the log record generated by this Autoencoder may not generate very similar log record in the system.

The difference between top-down/bottom-up and real solution for each parallel in each of the category of their logs. Thresholds per cat can be different because there may be different apps and network devices, different sec systems and they have other normal behaviours of what to rebuild. If the deviation is reconstructed from the records, and deviation was relatively large or small, all the records will be marked as 'potentially anomalous' and it will be submitted for analysis in a relational form. This process can detect anomalies even if there aren't many manually labelled anomaly records, or no supervision is involved. Reconstruction deviation is not an analysis signal, but is a sign of a security event or an expectation/refrain from complying with standards violation. Figure 3 outlines the reconstruction-based anomaly identification process, showing how deviations between original and reconstructed log features support anomaly assessment.

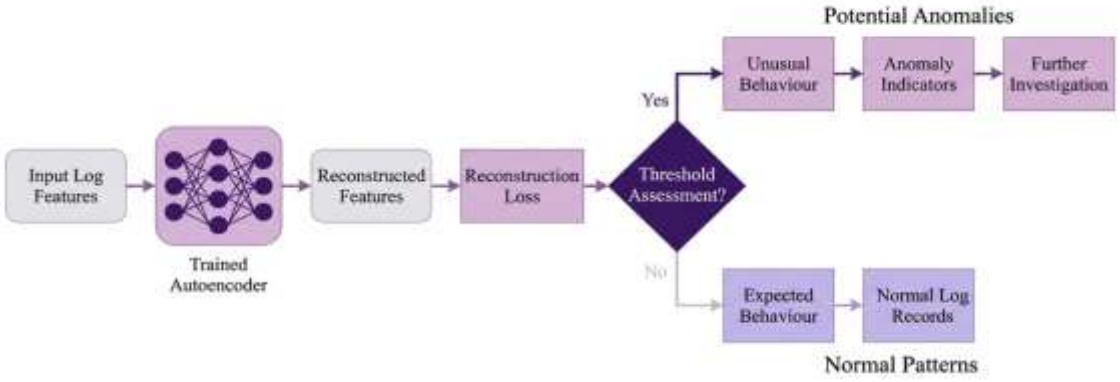


Figure 3. Reconstruction-Based Anomaly Identification Workflow

3.4. Cross-Domain Graph Construction and Relationship Mapping

The fourth step: It is transforming log representations learned, into graph-based representation. Changes in the log stream can be anomalies detected by an autoencoder, enterprise anomalies may consist of relationship anomalies between multiple systems and types of events. To explore all these kinds of dependency a graph representation can be used.

N is shown as nodes for proposed graph along with the edges. Nodes include log events, apps, databases, network devices, user accounts and security related objects. Relationships are described between relationships such as event relationships (successions), communication relationships (dependencies), shared identifier relationships, relationships between resources and operational relationships – these are represented as edges. Event category, time and reconstruction based anomaly indicators can be used to as node attributes of parallel autoencoders, aside from image frame (box) data.

One good thing about this graph is that it allows to say a relationship that may not be allowed by the log. For example, after a normal logon, events that include connection/communication with the databases could ensue. These can be represented as nodes which are connected and the analytical framework can be in the context of this combination. Therefore, generating graphs is something that involves matching features learnt from the various sources with operational dependencies in the enterprise. Figure 4 visualizes the heterogeneous enterprise graph, demonstrating how log events and infrastructure entities are connected through operational relationships.

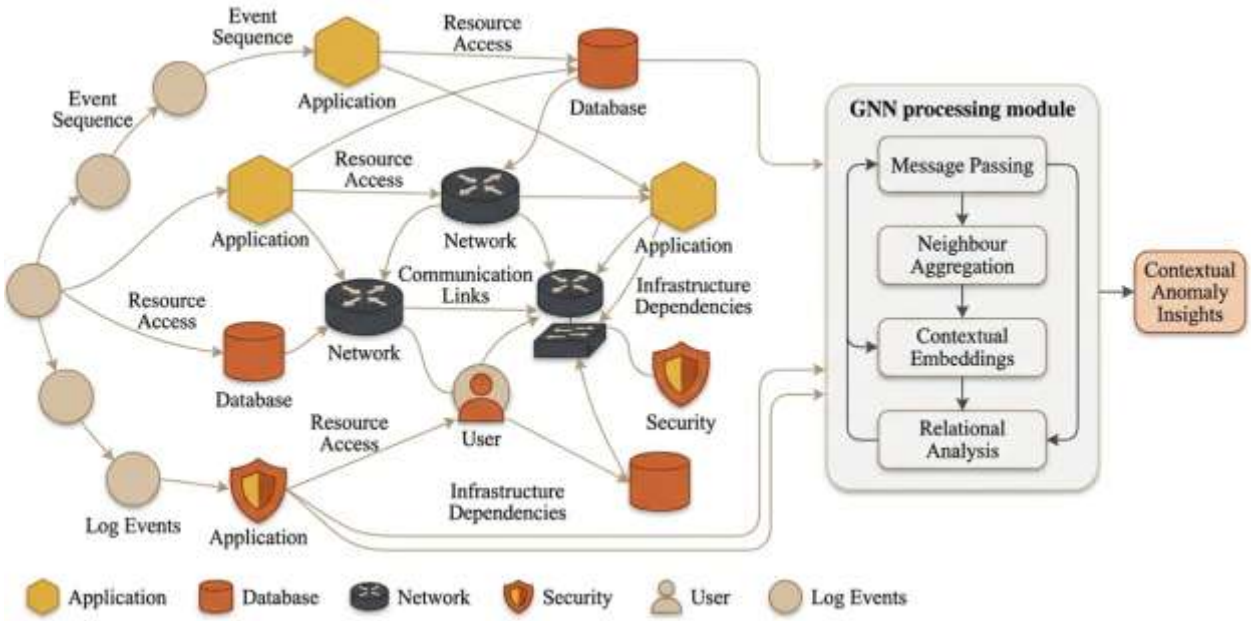


Figure 4. Heterogeneous Enterprise Log Graph and GNN-Based Relationship Mapping

3.5. GNN-Based Relational Intelligence and Anomaly Contextualization

The fifth stage, a series of Graph nerual Networks (GNNs) are used to learn a representation over the constructed heterogeneous log graph. Each node can provide information (via GNN) to its neighbors and each node can (also via GNN) generate a representation that is affected by its neighbors. Then, it can be aggregated from events, and then (cross domain) it can be aggregated by successive aggregation.

The representations of the nodes include the latent representation of the autoencoder and some information from the graph-neighbourhood. It allows for the framework to take into account that a certain event is not only “stands out” from the other similar events in the enterprise, but stands out from other similar activities as well. For instance, if the featured representation of a network event is just a user accessing an application, but is accompanied by other unusual authentication of the user's login account, then the event might have a larger impact. Likewise, if two or more logs away from the normal log reading form a trend in the logs, this is considered to be significant trend of operation.

GNN, however, can provide some context on the relational anomaly from the errors in isolation that are possible with others. It can also be used to discover sets of event groups, unknown dependency relationships between the groups and possibly the relevant group of events between the systems. Various different heterogeneous enterprise scenarios can have anomalies within the context of the graphs created.

3.6. Compliance-Risk Interpretation and Analytical Outputs

The last stage tries to map the output of the parallel autoencoders and the relationship analysis based on GNN to compliance-risk oriented information. Deviations from the reconstruction have to be reflected by the framework as well as graph connectivity and relation among events, source categories and operational meta data to facilitate detection and comprehension of potentially significant activities.

The analytical results can be presented as a source specific anomaly indicator, pattern of events that are interrelated, abnormal access relations or as descriptions and records of cross domain dependencies for visual review. These outputs can be useful for the enterprise monitoring, security analysis, operational audit and compliance analysis. Patterns (that have been identified) can be connected with your regulatory requirement or organisational policy (if you have one) to help with examining this evidence.

What's interesting is that not all of the deviations found have to be non-conforming acts, however. Instead, it provides contextual details that can aid in audits, security analyses and enterprise admins in prioritizing investigations and also manage evidence. They should be treated by human treatment of deviations identified, particularly in terms of the organisational context and interpretation of the significance of deviations. Figure 5 presents the compliance-oriented analytical layer, linking anomaly indicators and graph-derived context with evidence prioritization, auditor review, and compliance reporting.

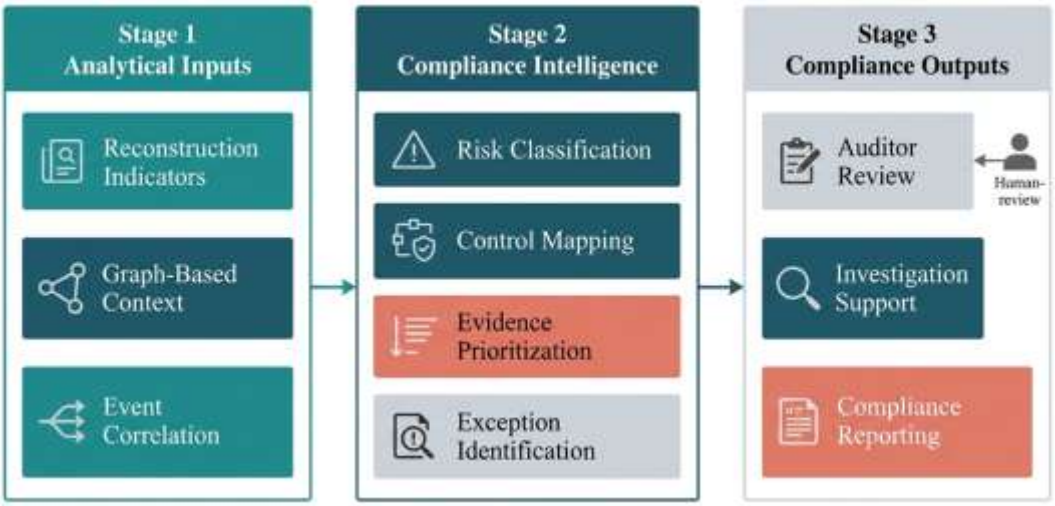


Figure 5. Compliance-Risk Interpretation and Analytical Output Architecture

4. Empirical Appraisal of Integrated Log Intelligence and Cross-Domain Analytical Capabilities

By combining with the Parallel Autoencoder–GNN framework, its architecture attributes, analysing capabilities and suitability to a heterogeneous enterprise can be evaluated. This appraisal takes into account the ability of its integrated components to tackle the known problems of log analysis, without introducing experimental measurements, or empirical validation of results from an implemented system.

4.1. Heterogeneous Data Representation

The parallel autoencoder architecture can be used for organization of the variability across four types of logs: application log files, database log files, network log files and security log files. Know how to use information in data sources of streams in order to create compact representations. Thus in the heterogeneous representation scheme these problems can't be addressed with a single representation of a model for operational records.

4.2. Anomaly Identification and Contextual Analysis

There are ways to detect violated learned log-patterns by a reconstruction based analysis. It can be applied along with record-level and global modelling using GNNs to offer capability of record-level and global anomaly interpretation. The framework can be used to perform a broader analysis, considering the relationships between interconnected events, that may not be considered normal enterprise activities.

4.3. Cross-Domain Relationship Mapping

A “part” with graph-based information has to be also mandatory “part” to be able to model Enterprise things too (e.g dependency relationship). User element, application element, database element and network element can have relationships established between them and can be examined in one model. This is useful when communicating the meaning of the linked activities which might be provided in the traditional monitoring solution based on the source(s) of the activities.

4.4. Compliance and Operational Relevance

Anomaly indicators and graphical context data can be utilized in the security monitoring or in the compliance risk assessment. The analytical results can be searched from the auditor and/or administrator while reviewing the events to determine the need for investigation and rank the events, as well as access events and anomalies for further investigation. There is no alternative to the human interpretation which is an important component in determining the regulatory significance.

4.5. Overall Framework Appraisal

The framework can be considered as additional addition of anomaly identification and relational graph intelligence as a whole framework (known as SSRL). This is mainly because it can link log attributes and their dependency from various domains. However, the effectiveness of its use would be highly dependent on the type and quality of logs it was based on, the accuracy required to draw the graphs, the setting of the threshold and the scalability and the living organism itself.

5. Discussion: Convergence of Parallel Representation Learning and Graph-Based Enterprise Intelligence

One of the other important components in the proposed solution is the integration of the knowledge of Logs representation learning in additional to Relational analysis for Enterprise monitoring. Each log source has its own parallel autoencoder which allows the log source properties of the log data (application, database, network and security logs) to be maintained throughout the analysis. The huge amount of labelled datasets not required, construction based anomaly identification also provides help in the anomaly detection. However individual anomaly detection cannot necessarily bring the context of operation which also includes correlated events.

This can be alleviated though, by modeling the connection between Enterprise entities and Log events in the form of Graph neural network. All this combined creates an easier method to establish a comprehensive view of the entire distributed operational activity, particularly where unforeseen activity may occur in various aspects of the infrastructure. It also enables investigation of dependency of events, and additional relationships of events with various contexts that are usually not captured by regular per source monitoring.

The framework will expand enterprise-governance to a broader scope of log analysis, compliance-risk interpretation. Anomaly indicators – which can be associated with graph relationships – offer context to provide security analysts and auditors with clues as to what's important and what evidence to focus on. But, when it comes to a real use case deployment, there are some issues to resolve towards consistency in data, quality of the graph, scalability in the computation, the value of threshold and explainability. Of particular importance is the architecture is not operationally available but rather it is an analytical architecture, an architecture that has not yet been proven. The direction based on parallel learning and graph intelligence will be a promising way to construct the “context-aware and heterogeneous” enterprise log management systems.

6. Conclusion

In this work, a conceptual framework which consists of parallel autoencoders along with Graph Neural Networks (GNNs) for enterprise log mapping is proposed for enterprise log information of many varieties. Issues that must be addressed in the solution with the suggested architecture involve enterprise components complexity, data type in each source system and different log format fragmentation. A parallel autoencoder can represent different types of logs to have a small sized vector, and the deviations from the operation patterns learnt by parallel autoencoder can be found by reconstruction based analysis. By adding support for GNNs, both the representation and analysis of relations between log events, applications, databases, network devices and security entities is now possible.

Combines the traditional log analysis with the "across domain relational intelligence" given by the source specific anomaly identification. Its solution's analytical layer is compliance-oriented too and can serve as an entry point to start identifying potentially related operations, and evidence-based auditing. The design does not consider any particular idiosyncrasy a per se failure – it is the focus on the human exploration and interpretation.

In conclusion, it is probably a "next step" for smart enterprise log management systems to have a parallel representation learning and graph analysis. Potential for future studies could involve implementation for practice, systematic validation, scalability study, explainable anomaly detection and incorporation with organizational compliance needs. The changes could also impact to scope the framework in an enterprise environment where it can be complex, distributed and very secure.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

[2] Du, M., Li, F., Zheng, G., & Srikumar, V. (2017). DeepLog: Anomaly detection and diagnosis from system logs through deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1285–1298). Association for Computing Machinery. https://doi.org/10.1145/3133956.3134015

[3] Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log anomaly detection via BERT. In 2021 International Joint Conference on Neural Networks (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN52387.2021.9534113

[4] He, P., Zhu, J., Zheng, Z., & Lyu, M. R. (2017). Drain: An online log parsing approach with fixed depth tree. In 2017 IEEE International Conference on Web Services (pp. 33–40). IEEE. https://doi.org/10.1109/ICWS.2017.13

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[6] Huang, S., Liu, Y., Fung, C., He, R., Zhao, Y., Yang, H., & Luan, Z. (2020). HitAnomaly: Hierarchical transformers for anomaly detection in system log. IEEE Transactions on Network and Service Management, 17(4), 2064–2076. https://doi.org/10.1109/TNSM.2020.3034647

[7] Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. Machine Learning with Applications, 12, Article 100470. https://doi.org/10.1016/j.mlwa.2023.100470

[8] Le, V.-H., & Zhang, H. (2021). Log-based anomaly detection without log parsing. In 2021 36th IEEE/ACM International Conference on Automated Software Engineering (pp. 492–504). IEEE. https://doi.org/10.1109/ASE51524.2021.9678773

[9] Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., & Sun, P. (2019). LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (pp. 4739–4745). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2019/658

[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv. https://doi.org/10.48550/arXiv.1301.3781

[11] Nedelkoski, S., Bogatinovski, J., Acker, A., Cardoso, J., & Kao, O. (2020). Self-attentive classification-based anomaly detection in unstructured logs. In 2020 IEEE International Conference on Data Mining (pp. 1196–1201). IEEE. https://doi.org/10.1109/ICDM50108.2020.00148

[12] Oliner, A. J., & Stearley, J. (2007). What supercomputers say: A study of five system logs. In Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (pp. 575–584). IEEE. https://doi.org/10.1109/DSN.2007.103

[13] Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

[14] Zhang, X., Xu, Y., Lin, Q., Qiao, B., Zhang, H., Dang, Y., Xie, C., Yang, X., Cheng, Q., Li, Z., & Chen, J. (2019). Robust log-based anomaly detection on unstable log data. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (pp. 807–817). Association for Computing Machinery. https://doi.org/10.1145/3338906.3338931