## 4.1 Dataset

There are totally 200,000 anomaly event records of our distributed systems gathered spanning a period of the whole year from early May 2016 to May 2017 are available here for our experimental studies. Using a larger set of data as the input for a deep learning model is not always the best choice, as this would increase the dimension of the parameters, and also introducing sparsity issues. As a result, it may negatively impact the performance of classification algorithms, hence we decide to take the half of the original dataset to carry out the experiments. To validate our results effectively and precisely, we have labeled and checked all dataset manually and assigning each of them a specified class. The examples of contents of log message, anomaly types and classification labels are shown in Table 1. To achieve more accurate classification performance, data preprocessing, feature generation namely data conversion and normalization need be taken into account before the formal training of models. The detailed procedures of log text preprocessing and numerical feature generation modules are depicted in Fig. 1.

Table 1. Categorization and Labeling of Anomaly Events

|  |  |  |
| --- | --- | --- |
| **Content of Log** | **Category** | **Label** |
| Could not find keytab file: /etc/libvirt/krb5 | File | 0 |
| No DHCPOFFERS received | Network | 1 |
| Wake up new task 0xd3/0x120 | Service | 2 |
| Error dropping database (can’t rmdir \testdb\) | Database | 3 |
| WARNING: Unable to determine session | Communication | 4 |
| Read-error on swap-device (253:1.16968112) | Memory | 5 |
| No CSI structure available | Driver | 6 |
| respawning too fast: disabled for 5 minutes | System | 7 |
| Security real capable no audit | Security | 10 |
| FAILED SMART self-check. BACK UP DATA NOW | Disk | 11 |
| Error: selected processor does not support ‘strexb’ | Processor | 12 |
| Buffer I/O error on device dev sda | IO | 8 |

**Preprocessing.** Irrelevant or redundant features normally bring much noise to the feature extraction tasks, and resulting in inferior classification performances, in order to generate appropriate features for models to handle with, a task like practical anomaly classification often requires a data preprocessing, which removes the noise from the raw data and leads to statistically significant increase in classification accuracy [2]. The step of the preprocessing procedure consists of filtering out redundant information through stop-words and punctuations. Our stop-words reference Long Stopword List1 and punctuations comes from all commonly used symbols in the English language.

**Feature Generation.** Previous work [25] proposed to discover the regular pattern of system log messages by tracking the source code, but it is not an effective way for handling with sophisticated distributed system logs. The original gathered dataset here is the unstructured log text, which is unable to be processed in the neural networks, consequently, construction of numerical event-wise feature sets representing the semantics of each log message field on real distributed systems for facilitating the calculation in the deep learning models has been conceived with the objective of fundamentally maintaining the underlying structural properties of the temporal log events as much as possible. Such formatted numerical feature layout allows us to learn and analyze the temporal events efficiently for anomaly categorization tasks, additionally, it protects the sensitive information with respect to system operations at the same time.

Feature Generation

Semantic Similarity

Feature Vector

Normalization

Training Dataset

Preprocessing

Filtering Stop-Words

Filtering Symbols

Raw Log Events

1. <http://www.ranks.nl/stopwords/>

Fig. 1 Flowchart of Data Preprocessing and Feature Generation

In reality, the structured numeric feature generation from an unstructured log message text requires a solid background knowledge of the original information, given that projected features are so different from the original patterns from a data view [12]. Therefore, we have built up a number of dictionary libraries involving total 14 topic types for numerical data transformation of log message events and enable them to extract important pattern features to implement numeric computation in our model architectures. Every topic type of dictionary libraries contains tens of keywords to represent the characteristics of different anomalous log events, which then would be labeled properly with specific numbers. For every message sequence of anomaly log events, the semantic similarity value between them and all keywords in the dictionary libraries is regarded as a generated feature vector. The numeric features gained in this way not only retain certain textual structures of the original log events, but also maintain the differentiation among a variety of anomaly types. The semantic similarity of anomaly events and dictionary keywords is defined by:

1. <http://www.ranks.nl/stopwords/>

(1)

where refers to Levenshtein distance [34], which is also referred to as edit distance [35], a string metric for measuring the difference between sequence A and sequence B (Here, A refers to log message sequence, B refers to keyword sequence of dictionary libraries). Note that the first element in the Equation 1 corresponds to the primitive semantic similarity equation, for ease of calculation, it is multiplied by 10.

Considering the time consuming nature of the feature construction and feature projection in deep CNN training tasks, we merely select top 10 semantic similarity values between each single log event and all topic types in dictionary libraries, then combine them into a feature vector as the representative of this anomaly log event. As a consequence, each event record contains 140 numerical event-wise attributes. Furthermore, normalization, to the best of our knowledge, is the final step before the anomaly classification algorithm being executed. The min-max normalization which usually normalizes every feature into a [0, 1] interval is commonly used in practical applications, it is also employed in our experiments. Ultimately, normalized numerical features and matching labels are utilized as dataset to evaluate our proposed models.

## Reference

[2] Smith M. R., Martinez T. Improving classification accuracy by identifying and removing instances that should be misclassified. The 2011 International Joint Conference on Neural Networks. 2011; p. 2690. ISBN 978-1-4244-9635-8. doi:10.1109/IJCNN.2011.6033571.

[12] Goldstein M. In: Markus Hofmann RK, editor. Anomaly Detection. Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC. 2013; p.367-394.

[25] R. Vaarandi. A data clustering algorithm for mining patterns from event logs,” in In Proceedings of the 2003 IEEE Workshop on IP Operations and Management (IPOM), 2013; pp. 118-126.

[34] Levenshtein VI. Binary codes capable of correcting deletions, insertions and reversals. Journal of Soviet Physics Doklady, 1966, 10:707-711.

[35] Navarro, Gonzalo (1 March 2001). A guided tour to approximate string matching. ACM Computing Surveys. 33 (1): 31-88. doi:10.1145/375360.375365. Retrieved 19 March 2015.