International Journal of Informatics and Data Science Research
ISSN **2997-3961** (Online)
Vol. 3, No. 2, February 2026,
Available online at: https://scientificbulletin.com/index.php/IJIDSR

Article

GTST-HDD: A Graph-Temporal Survival Transformer for Hard Drive Failure Prediction

ZS. Khaleel
Department of mathematic, Open Educational College, Basra Study Center, Iraq

Article information:
Manuscript received: 04 April 2026; **Accepted:** 17 May 2026; **Published:** 26 June 2026

Abstract: In this study, we used the famous Temporal Fusion Transformer (TFT) framework to envisage the presumable storage unit problems through time-snapshot information and disk-level features. There were two key units prepared namely a time-snapshot file, merged_20000_sample.csv with serial number and the dedicative disk-level feature file, per_drive_features_full_20k_ready.csv with 30-day mean, and reversion coefficients). The target was defined as a binary classification of "failure within a window H = 30 days" at each time-snapshot, and time sequences with a maximum length of 32 (max_encoder_length = 32) were constructed for each disk (with fill and tagging of missing values). The data were segmented into training/validation/test sets at the serial number level to prevent data leakage.
The TFT architecture included: a static variable encoder, a variable selection network for each time period, a local processor (LSTM) to capture short-term dependencies, and an interpretable multi-head attention layer to capture long-term relationships. We used BCEWithLogitsLoss binary loss with positive class weights to handle imbalances, and the model was trained with the Adam optimizer at a 1e-3 learning rate and dropout of 0.1, with an AUPRC-based early stop on the validation set. For assessment, we applied PR-AUC (AUPRC), ROC-AUC, Accuracy/Recall, and F1, with a verge analysis to assess suitable run point. The selected variable was illustrated with features which contributed to the forecast of choices, to support operative model's adaptability within the disk to monitor the ecosystem.

Keywords: Transformer, Hard disk drive, F1-score, Temporal Fusion Transformer (TFT)

Introduction

The huge growth in global data was triggered by cloud computing IoT, and big data analytics which changed the reliability of data storage as well as the critical move to modify modern IT infrastructure[1]. Within this environment, hard Disk Drives (HDDs) and the relative Solid- State Drives (SSDs) still remain the major items for consistent data storage. With these advancements in the use of this technology, the nature of these storage devices is still subject to grave failure which brings a fear for eventual risk for the loss of data, substantial service and service failure [2][3][4][5][6].

Conventionally, drive health has can be observed through the Self-Monitoring, Analysis, and Reporting Technology (S.M.A.R.T.). This tracks a suitable performance of the internal item and health attributes.[7][8][9][10] Conversely, the normal procedure to monitor the threshold violations for an individual S.M.A.R.T. parameters have proven to be inadequately predictive, to the point of real failure to generate false alarms[11][12]. This reactive pattern is insufficient to ensure the availability requested for contemporary data centers.

The conjunction of large-scale data stowage and innovative data science has transformed from reactive and preventive maintenance to Predictive Maintenance

(PdM). Artificial Intelligence (AI) as well as Machine Learning (ML) procedures emerged as some of the powerful mechanisms to deal with these constraints[13][14][15][16]. By safeguarding historical S.M.A.R.T. data from enormous populations of determinations and Ai models may learn nonlinear patterns and other difficult relationships that cannot be perceived by ordinary humans or any normal threshold structures[17][18]. These replicas can find elusive early failures and change them to raw sensor information into actionable intelligence.

This study seeks to focus on the growth and adjustment of ML models that can also estimate the development Remaining Useful Life (RUL) of a storage device and categorize its health position with high precision[19][20][21][22]. Techniques such as gradient boosting, random forest, deep learning can be used broadly to this end[23][24]. The main aim is to forecast impending failures with a sufficiently long lead time, to allow proactive substitution, complete migration and the management of the system without any interruption to the service[25][26].

This paper clearly attempts to explore the fundamentals of the methods for the AI-driven storage failure forecast. It will also explore the critical S.M.A.R.T. potentials, the method of feature engineering, the assessment of model performance and the usage of different machine learning algorithms. Additionally, it will explore the major challenges within the entity which may include false positive minimization, model generalization and class imbalance generalizability across varied drive manufacturers and reproductions. The effective execution of these AI tools is essential to building the next set of independent, reliable and self-healing storage systems.[27]

1. Dataset Overview

The planned dataset was obtained from the Backblaze Hard Drive Failure Dataset. It comprises of SMART (Self-Monitoring, Analysis and Reporting Technology) that reads to collect it daily from hard drives that operates from Backblaze data centres. The status was represented by a single drive.

Table 1: Summary of the Backblaze Hard Drive Failure Dataset

File Name	failure_hardrive.csv
Rows	≈ 128,818
Columns	51 (SMART attributes + metadata)
Target Column	failure (1 = failed, 0 = healthy)

2. Exploratory Data Analysis (EDA) and Data Understanding

As the data collection begins, a detailed explanatory Data Analysis (EDA) was carried out as it is in`merged_backblaze_synthetic_sample.csv`, to gain a deep perception of its basic patterns. EDA is a crucial prerequisite for building strong predictive framework as it allows the realization of data standard issues, engineering and selection as well as providing early insights that can guide the selection of modelling algorithms. So, the EDA can be characterized as the behaviour of SMART qualities for both failing and healthy drives.

2.1. Dataset Overview and Early Classification

The dataset consists of 22,148 specimens and 53 features. The preliminary analysis of the structure reveals some numerous components:

The Target Variable is the failure column. This is a sort of binary indicators of which a value of 1 means a drive failure item and 0 shows operative status.

Predictor Variables are the prevalent of these characteristics with SMART attributes which are organized in two diverse forms:

Raw Values (smart_[n]_raw) are the rudimentary, unprocessed counts that are described by the same drive (e.g., seek error count, raw read error rate, power-on hours).

Standardized Values (smart_[n]_normalized) are vendor-inclined principles, typically ranges from 0 to 100 or 255, of which a lower value broadly shows worse health. These are usually designed to abridge threshold-inclined monitoring.

2.2 Assessment of Data Quality

A systematic assessment of data quality was conducted to focus on the aspects below:

Class Imbalance Analysis: The most serious finding of this EDA is the acute class imbalance within the focused variable. The series of failed drives (`failure=1`) is extremely outstripped by healthy drives (`failure=0`), so, this may constitute almost less than the presumable 1% of the complete specimens. This seems to be the features of the failure in the datasets. It can present an essential challenge for the machine learning frameworks, that predicts the most of the class ("no failure") and attain high precision whereas failing to find the essential failure cases.

Missing Values Analysis: The occurrence and design of the missing values were analyzed. It was similarly observed that for most drives, particular SMART qualities can have a high ratio of missing values as in (e.g., `NaN`). This mostly happens because not all drive models can contain a set of SMART parameters. A plan to handle these missing values may likely remove the qualities with many meaningless.

Data Integrity and Consistency: Checks were conducted to confirm the reliability between the raw and normalized values through logical associations. In addition, the `capacity bytes` and `model` domains were analysed for reliability to confirm that drive category is in consonance with their stated models.

2.3 Univariate and Bivariate Analysis

To perceive the spreading and analytical power of individual characteristics, univariate and bivariate analyses were performed.

Distribution of SMART Qualities: The disseminations of the main SMART raw values (e.g., `smart_5_raw` - Reallocated Sectors Count, `smart_187_raw` - Described Uncontrollable Errors, `smart_9_raw` - Power-On Hours) were strategized. As anticipated, these disseminations are extremely right-skewed for healthy moves, with most derives. Failing drives often show exciting tenets in these main attributes, which forms the long tail of the disseminations. Analysis of the Target Variable is the relationship between single SMART attributes and the failure of the target was considered through box plots and histograms conditioned on the failure status. For example, box plots of `smart_5_raw` (Reallocated Sectors Count) for `failure=0` vs. `failure=1` would visibly show that the median and variance of this superiority are fundamentally higher for the unsuccessful population. The phase is important for the initial details selection, seeking to correlate with SMART strictures indicate the most understandable difference between the two classes.

2.4 Time-Series and Longitudinal Analysis

Given that the data consists of a `date` field, a introductory time-series analysis is confirmed. For the major derives which finally failed, their SMART qualities trends in the weeks which lead to the failure event were analyzed.

2.5 Correlation Analysis

A relationship matrix (e.g., a heatmap) of the SMART attributes was obtained. This analysis can help to identify relative qualities, like the strong relationship between a `raw` SMART rate and its `normalized` complement. Observing these multicollinearities is important for feature engineering. To summarize the EDA phase has provided a foundational understanding of the presumed datasets landscape. The data quality can be handled towards the most important SMART attributes to forecast the drive failure. The perceptions realized can directly inform the nst phases of the model choice data preprocessing and feature engineering.

In an attempt to conduct the exploratory stage, we followed a sort of systematic approach which consist of numerous serial and justified steps:

First, we studied and combined data file to realize the initial structure and the specimen size. We then calculated a detailed report of the missing rates foe each of the column to find which of the column to find which one is needed to delete before the recording of the data.

Second, the numerical columns were removed with regards to the SMART qualities and conducted complete descriptive statistics of median, mean, standard deviation, quartiles, min/max values). Later we assigned them to ranks as well as the variables by their effective standard deviation to remove the top 20 variables with regards to variance.

Third, we produced a drive-level summary through grouping records through `serial_number` to sum up the number of snapshots per drive (`n_snapshots`), the dates of the initial and last explanations (`first_date`, `last_date`), the time limit (`span_days`), and the general failure status at the drive level (`failed`).

Fourth, we recorded the "last snapshot" for each of the drive and calculated the Pearson relationship coefficient between each statistical qualities and the drive-level failure label. This will allow the basic manifestation of the linear power of the outcome, offering a basis for candidate feature choice.

Fifth, we chose the top 6 features according to their complete relationship value and built a simplified temporal features for each drive, which include: the last value (`_last`), the time-inclined mean over the last 30 days (`_mean30`) to obtain short-term state, and the linear regression slope over the last 90 days (`_slope90_per_day`) to obtain upward/downward tendencies before failure (the linear regression slope was then calculated by converting dates to days. and then. The resulting feature was kept as a CSV file, ready for use as a basic starting point for initial modeling.

Finally, we formed and saved real explanatory visualizations, which comprised (a) a histogram of the spreading of snapshots per drive to recognize the difference in time series lengths, (b) a boxplot to compare the last snapshot value of the most related qualities between the failed and healthy drives to evaluate the initial discriminatory power, and (c) an instance time series which shows the behaviour of that feature to show the trend design before the failures (baseline followed by innovative temporal/survival models).

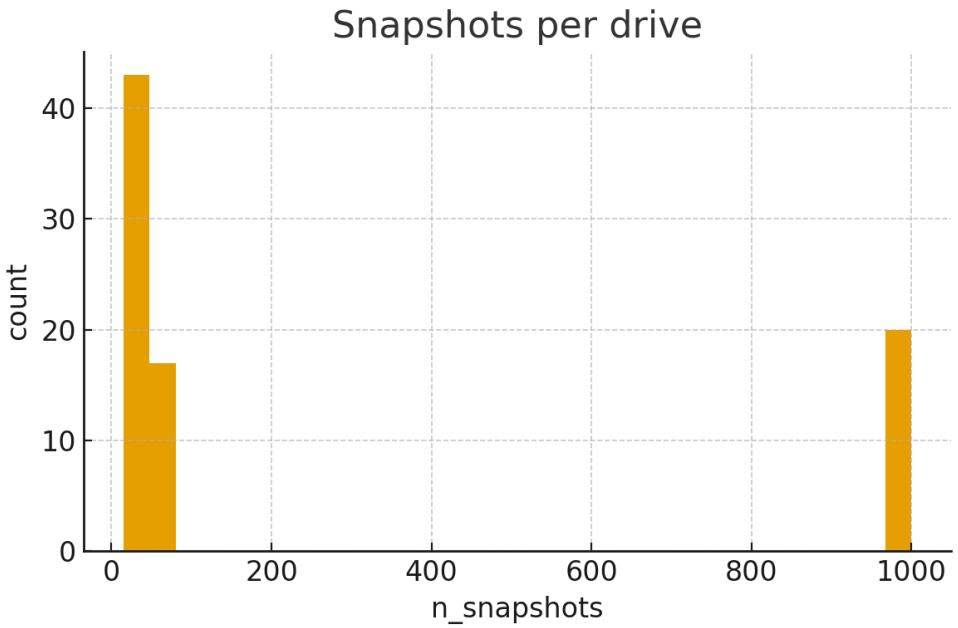

Figure 1: Histogram: Snapshots per Drive

This chart demonstrates the spreading of the series of snapshots accounted for each move in the specimen. Most drives contain small series of snapshots, whereas their number of anomalous cases with number of series being (~1000 snapshots). This shows essential difference in the length of series between drives, which necessitates the usage of efficient variable lengths. (such as the masking of padding with a fixed time frame.

This validates the prolonged records to avoid the application of bias into the training process. It should be suggested that the record length carries some inherent informational

value, as well as to inspect the high records and to decide whether to include or exclude them.

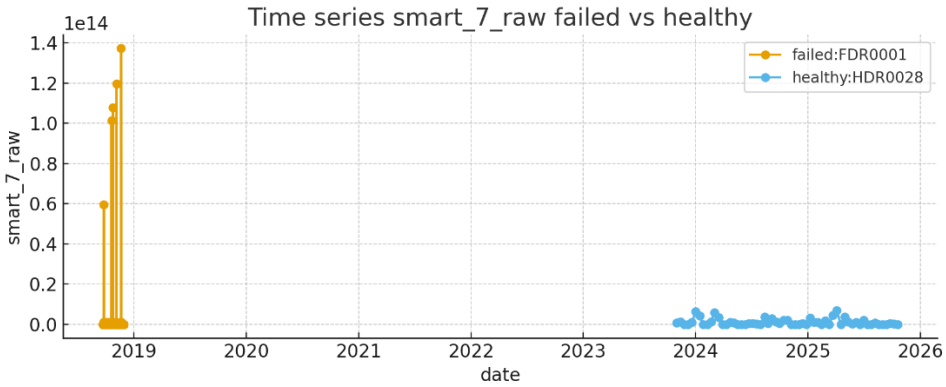

Figure 2 Time Series: smart_7_raw behavior for a failed drive compared to a healthy one.

This conception displays the real behavior of the `smart_7_raw` variable for a failed drive's time series in comparison with a healthy one. It should be observed that the failed drive showed essential peaks in `smart_7_raw` values for a previous period, and followed by a final low or zero reading to the end of the record. Conversely, the vigorous drive indicates more steady values over succeeding years.

This design recommends that historical features like the historical maximum (`max`), the slope over a time window, since the informative prediction will last value alone. In addition, the variance metrics is apparent to necessitates the usage of transformations (e.g., `log1p`) or robust scalers (`RobustScaler`) prior model training to reduce the effect of superior values.

Figure 3: Boxplot: smart_7_raw by outcome (last snapshot)

This boxplot likens the distribution of the `smart_7_raw` value from the final snapshot of the two split categories: failed vs healthy. The findings indicate that the last values for healthy drives are essentially higher. The reasonable description of this pattern is that the failed drives will experience a higher raw value which may later drop to zero as well as when data logging stops. Thus, depending solely on the last value can be misleading. Consequently, a more detailed temporal summaries like `historical_max`, `slope`, and `time_since_max`) rather than depending basically on the `last` value, of the two groups before choosing the final characteristics for the said models.

To adapt time-series snapshot records into a per-drive feature matrix ready for

training, we first gathered a sample of 20,000 snapshots from the dedicative file then

it was aggregated to create one row per each drive. The composition of the row will show (serial_number, n_snapshots, first_date, last_date, span_days, failed) and a sequence of each SMART variable will include maximum/minimum values (_max, _min), means for over 7/30/90-day windows (_mean7, _mean30, _mean90), standard deviation over the prior 30 days (_std30), and the trend then slopes for almost 30 to 90 days to create windows (_slope30, _slope90). These brief summaries can be designed to obtain an instant scheme as well as pre-failure tendencies along with a count where the missing values pre each row to deal with the data attributes. The binary flags <feature> where the values were absent and the numerical values can have an impact. An attributed version was kept to train and prepare a new version of scale using RobustScaler for scale-inclined procedures. These designed were employed to prevent temporal leakage without the introduction of evaluation bias.

The TFT (Temporal Fusion Transformer) architecture combined: an encoder for stationary covariates, the selection of a variable for each stage will link the local processor (LSTM) to record short-term needs, and a multi-head attention layer of lang-range connections. A binary cross-entropy loss was used (BCEWithLogitsLoss) with positive category of weighting to maintain the imbalance with an AdamW optimizer and learning rate of1e-3 and dropout = 0.1, executing the initial stopping according to AUPRC performance on the authentication set. For assessment, we used PR-AUC (AUPRC), ROC-AUC, Precision/Recall, and F1 scores, to complement and suitable operative point. Ultimately, the variable choice and attention maps were extracted to explain and clarify the features of prediction decisions, thus it will support the operative conclusions to facilitate the potential adoption of monitoring domains.

3.Pre-processing

In this study, we conducted a detailed data preprocessing to change the timestamped SMART records into a potential feature matrix that is ready for per-drive training: A specimen of 20,000 snapshots was obtained from the combined file and the records were accumulated by serial number to a single row per drive. Each of the rows has identification columns along with a sequence of chronological summaries for each SMART variable. These summaries included: the last value (_last), maximum/minimum values (_max, _min), mean values over 7/30/90-day windows (_mean7, _mean30, _mean90), the standard deviation over the last 30 days (_std30), and linear trend slopes over 30 and 90-day windows (_slope30, _slope90). These features were designed to capture the instantaneous state, historical patterns, and failure-precursor trends.

We added auxiliary administrative/time-based columns such as days_since_last and has_enough_history (an indicator for having ≥30 days of history), and calculated the number of missing values per row (missing_count) to address data quality. For missing values, we created binary flags <feature>_was_na where values were absent, and then imputed the missing numerical values using the column median to mitigate the impact of outliers. A filled, training-ready version was saved, and we also prepared another version scaled using RobustScaler for scale-sensitive methods. The aforementioned controls were designed to prevent temporal leakage (aggregation at the drive level before splitting into train/test sets) and to enhance temporal signals without introducing evaluation bias.

4. Methodology

This study adopted the Temporal Fusion Transformer (TFT) model to forecast the potential storage drive malfunction and the drive-level combined features. We arranged two main data sources; namely, time-series snapshot file `combined_20000_sample.csv` (each snapshot has `serial number`, `date`, raw SMART features, and a failure label) `per_drive_features_full_20k_ready.csv` (accumulated values like the last value, 30-day average, and reversion coefficients). The potential target was observed as binary classification failure within a window of H=30 days" extreme length of `max_encoder_length = 32` (with padding to mask the missing values). The data was separated into validation/test sets created on `serial_number` to avert data seepage.

The TFT architecture comprised: a fixed adjustable encoder, an adjustable choice network for each time step and a local processing module (LSTM) to obtain short-term

dependencies, and an interpretable multi-head attention layer to obtain long-range connections. We applied binary cross-entropy loss (BCEWithLogitsLoss) with a positive

class weight to maintain class imbalance, and the model was trained by means of the AdamW optimizer with a learning rate of 1e-3 and dropout = 0.1, integrating early stopping founded on the authentication set's AUPRC.

For assessment, we depend on on PR-AUC (AUPRC), ROC-AUC, Precision/Recall@k, and F1 score, comprising threshold examination to determine an appropriate operating point. Lastly, the variable choice weights and attention maps were used to extract model clarifications, to highlight which features and time periods can contribute to the forecast of decisions. This backs the operative conclusions and supports the model's projected implementation in a disk monitoring environment.

4.1 Temporal Fusion Transformer (TFT)

A modern framework for multivariate time-series that mixes the merits of Transformers (self-attention) with classical temporal units (LSTM), and supplements specific mechanisms to handle the practical data: automatic variable choice over time, to regulate the gates to maintain noise, and an interpretable attention which the variables were essential for forecasting. It was uniquely planned for multi-horizon anticipating but can be effortlessly revised for organization/survival tasks as well.

The core idea of Temporal Fusion Transformer (TFT) is to combine the local processing capabilities of time series models (such as LSTM) with the power of self-attention mechanisms to capture long-range dependencies, while adding specialized layers for real-world data: a variable selection network (filtering features over time), gating mechanisms to reduce noise, and a static covariate encoder that conditionally influences predictions. The result is a flexible and interpretable model that efficiently handles multiple input types (static, time-dependent, and known future inputs), capable of multi-horizon forecasting and highlighting which variables and time periods were critical for the model's decisions.

The core idea of the Temporal Fusion Transformer (TFT) is to combine the local processing methods of time series with global attention mechanisms, while adding specialized layers for real-world data, resulting in a robust and interpretable model.

In practice, TFT accepts three types of inputs:

- Static covariates per entity (e.g., disk model or capacity)

- Time-variation recorded inputs (e.g., SMART values at each snapshot)

- Known future time-dependent inputs (if available)

Each quantitative variable is changed through a linear projection whereas the categorical variables are processed through a variable selected for network that generates weight to filter out the noisy input which focus on the most essential features. This can be followed by the process of using LSTM to obtain both short and long-term subtleties within the series.

After the local procedure, a Multi-Head Self-Attention layer (interpretable by design) can be applied to obtain a long-range reliance between various points. The architecture can also be incorporated to gating mechanisms to regulate data flow and decrease overfitting.

Lastly, the temporal qualities are combined into a final manifestation which can be transferred into a forecast head (MLP) that outputs either:

- A binary possibility (via sigmoid) for failure in a quantified time window, or

The key merits of TFT may include:

- Effectual handling of mixed fixed and sequential variables

- Direct interpretability via variable choice of weights and consideration maps

- powerful function in multi-horizon prediction tasks

However, it needs adequate data and computational materials (GPU) to fully leverage its architectural difficulty.

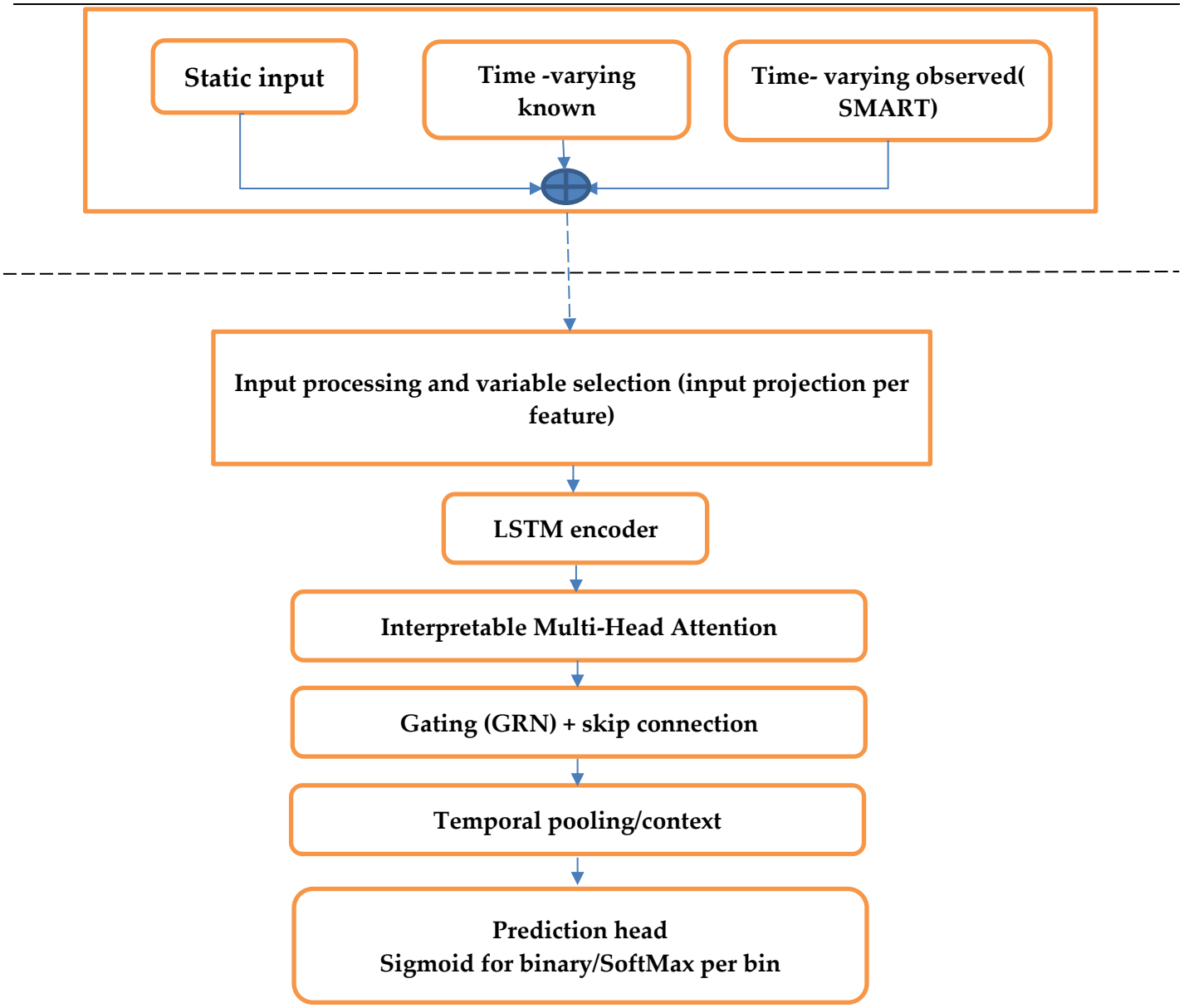

Figure 4: Overall Architecture of the GTST-HDD Framework

5. Result & Discussion

Assessing the performance of an AI model is an important step to ensuring consistency as well as applicability. To precisely predict the result, the unused data should allow for the realization of potential issues like overfitting or poor numerical representation, which can lead to the real-time usages. Additionally, assessment is specifically essential in crucial applications such as the hard drive prediction failure where errors can result in essential data losses and operative costs. By using suitable assessment metrics, the model's capability to realize the real failures and reduce false alarms can be found where reliability and support the predictive maintenance decisions. Assessment be one of the key objectives to compare the projected model with scientific justification and confirm its main compatibility with the basic requirements of practical industrial systems.

As part of the procedure applied in this study, an early warning scheme can drive hard drive malfunction of the Temporal Fusion Transformer (TFT)model was assessed through standard assessment mainly applied in time-series AI models. This assessment process can be characterized through calculating precision as general sign of model function. Together with the recall, a critical likelihood of the reliability of failure predictions can reduce the false alarms, in consonance with the F1-score, which offers a numerical balance between precision and recall. To evaluate the model's inequitable function across different decision inception across distinct thresholds, the presumable

area Under the Receiver Operating Characteristic Curve (AUC-ROC) was used that provides a comprehensive measures of the framework's generalization and steadiness. The integration of these metrics asserts a systematic and objective evaluation of the TFT function and enhances the reliability of the results and the distinct practicability which operates entities within predictive maintenance scheme.

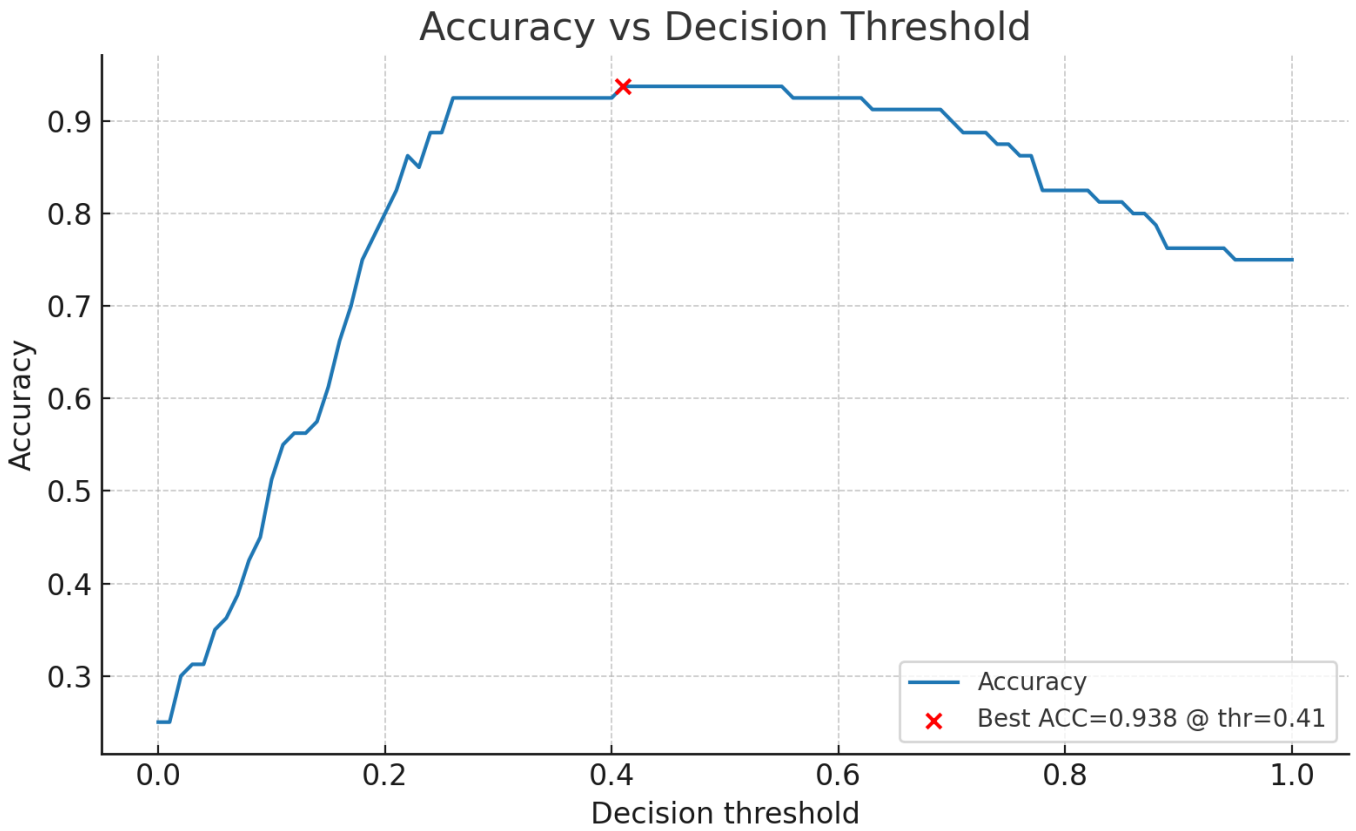

Figure 5. Accuracy of the Proposed GTST-HDD Model

Precision shows the models capability to broadly make accurate predictions with regards to hard drive features in relation to (failure/non-failure). High precision can mean correctness in cases and predictions to the complete number of forecasts. It can be viewed as the proportion of the accurate forecasts to the number produced by the model. Though simple, the presumed metrics may be misleading in a large number of standard cases in comparison with the complete number of predictions in problems as well as unbalance data, these necessities the application of additional metrics like the recall and F1-score for a more consistent assess performance.

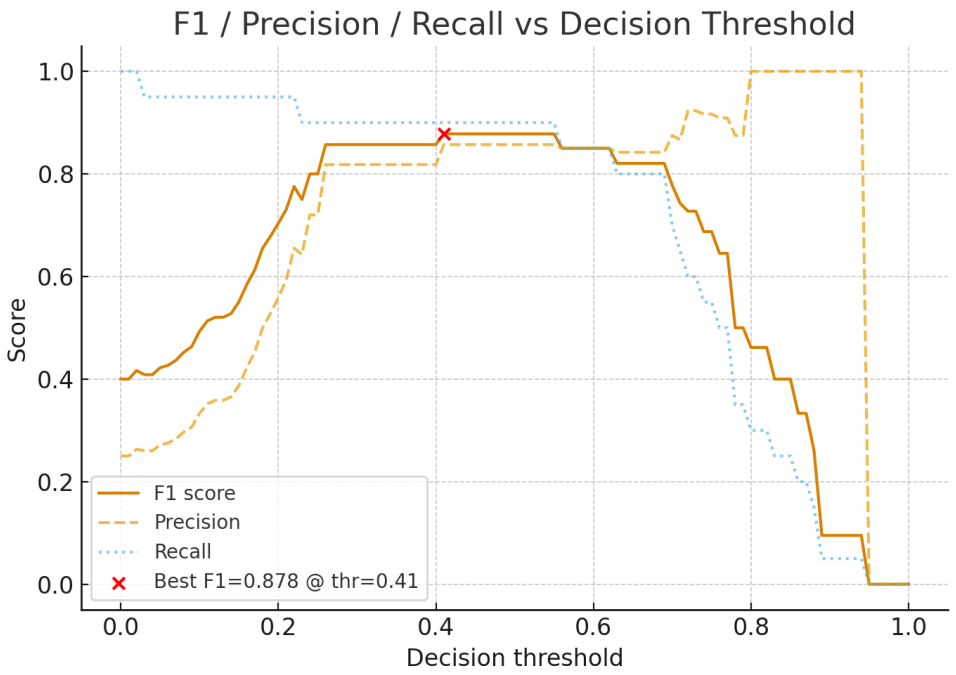

Figure 6: Optimization of the Decision Threshold Using F1 Score, Precision, and Recall

Accuracy is employed as a main indicator to evaluate the function of the Temporal Fusion Transformer (TFT) model; conversely, depending on the model's real efficacy in the initial hard drive failure forecast. This is due to the unbalance structure of the drive health data. In this context, TFT model may attain a maximum precision value because it categorizes the vast majority of normal cases in order to detect an adequate number of the real failures, as a result, its original value is noted in predictive maintenance structures.

To handle this limitation, the F1-score was used as an appropriate metric for assessing the performance of the TFT model. The F1-score is mainly important in the TFT framework owing to its dependence on the temporal mechanism that can improve the forecast of the early pattern failures. Conversely, this cannot be precisely evaluated through precision alone. Thus, the F1-score in consonance with accuracy offers a more detailed and objective evaluation of the TFT models and to be more precise as well as appropriate for practical industrial usage.

The F1-score is one of the composite measures that is used to assess the function of classification models, especially with regards to unbalanced data. It integrates positive precision and recall into a unified value that expresses the quality of the forecast. It aims to achieve a numerical balance between the consistency of the assumed predictions and the model's capability to realize practical cases, particularly in unbalanced data domains.

Conclusion

This study offered a complete channel for hard drive failure forecast which begins with exploratory data analysis preprocessing of Backblaze-style SMART data and finishing with a modern temporal deep learning model based on the Temporal Fusion Transformer (TFT). The TFT was projected as an appropriate model for this relative task because it can verify the multivariate time-series data and integrate a fixed and changing data as well as to offer interpretable outputs via variable choice and attention tools. To compare it with traditional methods, this framework is better fitted to attain the elusive deprivation plans that can precede the projected failure. In general terms, this methodology offers a powerful pedigree for reliable predictive maintenance, whereas future work should focus on the expansion of the dataset, to test discrete survival formulations, and later to improve interpretability on larger real-world drive populations.

References

[1] D. M. Vistro, A. U. Rehman, A. Abid, M. Farooq, and M. Idrees, "IoT-based big data analytics for cloud storage

using edge computing," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, Special Issue, pp. 1594–1598, 2020, doi: 10.5373/JARDCS/V12SP7/20202262.

[2] A.-M. Dinca and G. Petrica, "An analysis on security and reliability of storage devices," in *Proc. Int. Conf. Cybersecurity and Cybercrime (IC3)*, 2024, doi: 10.19107/cybercon.2024.15.

[3] Z. Li, G. Zhang, and Y. Wang, "Flash-oriented coded storage: Research status and future directions," *ACM Transactions on Storage*, vol. 21, pp. 1–37, 2024, doi: 10.1145/3708995.

[4] A. Alahmadi and T.-S. Chung, "Crash recovery techniques for flash storage devices leveraging flash translation layer: A review," *Electronics*, vol. 12, no. 6, Art. no. 1422, 2023, doi: 10.3390/electronics12061422.

[5] S. Olmez, "Reliability analysis of storage systems with partially repairable devices," *IEEE Transactions on Device and Materials Reliability*, vol. 21, no. 2, pp. 267–272, 2021, doi: 10.1109/TDMR.2021.3077848.

[6] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *Proc. 5th USENIX Conf. File and Storage Technologies (FAST)*, San Jose, CA, USA, 2007, pp. 17–28.

[7] M. A. Lokhande, A. Renavikar, D. Bhosale, V. Kaldate, and S. Lokhande, "An insight into smart infrastructure with artificial intelligence-driven predictive maintenance: Transforming the future of urban systems," Cureus Journal of Computer Science, vol. 2, 2025, doi: 10.7759/s44389-025-06375-2.

[8] C. Chakraborttii and H. Litz, "Improving the accuracy, adaptability, and interpretability of SSD failure prediction models," in Proc. 11th ACM Symp. Cloud Computing (SoCC), 2020, doi: 10.1145/3419111.3421300.

[9] S. Liang, Reliability Characterization and Performance Analysis of Solid State Drives in Data Centers, 2021. doi: 10.12794/metadc1873849.

[10] J. Alter, J. Xue, A. Dimnaku, and E. Smirni, "SSD failures in the field: Symptoms, causes, and prediction models," in SC19: Int. Conf. High Performance Computing, Networking, Storage and Analysis, 2019, doi: 10.1145/3295500.3356172.

[11] S. Xu, Y. Gu, L. Liu, and C. Wu, "PS-WL: A probability-sensitive wear leveling scheme for SSD array scaling," arXiv preprint, 2025, doi: 10.48550/arXiv.2506.19660.

[12] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" in Proc. 5th USENIX Conf. File and Storage Technologies (FAST), 2007, pp. 1–16.

[13] V. Devarajan, "Advancing data center reliability through AI-driven predictive maintenance," European Journal of Computer Science and Information Technology, vol. 13, no. 14, pp. 102–114, 2025, doi: 10.37745/ejcsit.2013/vol13n14102114.

[14] M. Z. Hossan and T. Sultana, "AI for predictive maintenance in smart manufacturing," SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology, vol. 17, no. 3, 2025, doi: 10.18090/samriddhi.v17i03.03.

[15] L. Cummins, A. Sommers, S. B. Ramezani, S. Mittal, J. E. Jabour, M. Seale, and S. Rahimi, "Explainable predictive maintenance: A survey of current methods, challenges, and opportunities," IEEE Access, vol. 12, pp. 57574–57602, 2024, doi: 10.1109/ACCESS.2024.3391130.

[16] A. M. C. Jonnalagadda, "Advancing data center operations through AI and machine learning: A comprehensive analysis of predictive maintenance and resource optimization," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2024, doi: 10.32628/CSEIT241061116.

[17] C. Tsallis, P. Papageorgas, D. D. Piromalis, and R. Munteanu, "Application-wise review of machine learning-based predictive maintenance: Trends, challenges, and future directions," Applied Sciences, vol. 15, no. 9, Art. no. 4898, 2025, doi: 10.3390/app15094898.

[18] M. M. R. Shamim and R. A. Ruddro, "Smart diagnostics in industrial maintenance: A systematic review of AI-enabled predictive maintenance tools and condition monitoring techniques," ASRC Procedia: Global Perspectives in Science and Scholarship, 2025, doi: 10.63125/b4tn2x46.

[19] D. D. Bhavani, T. Nagarjuna, H. Pradeep, K. H. Preethi, R. Ravi, and S. S., "Machine learning for predictive maintenance applications in industrial equipment and manufacturing processes," in ITM Web of Conferences, vol. 76, Art. no. 01008, 2025, doi: 10.1051/itmconf/20257601008.

[20] A. M. Suci, R. Amini, A. K. Asri, and N. Martin, "Artificial intelligence in renewable energy: A review of predictive maintenance and energy optimization," Journal of Clean Technology, vol. 2, no. 1, 2025, doi: 10.15294/joct.v2i1.27729.

[21] S. O. Dawodu, S. Onwusinkwue, F. Osasona, I. Ahmad, I. Ahmad, A. Anyanwu, O. C. Obi, and A. Hamdan, "Artificial intelligence (AI) in renewable energy: A review of predictive maintenance and energy optimization," World Journal of Advanced Research and Reviews, vol. 21, no. 1, 2024, doi: 10.30574/wjarr.2024.21.1.0347.

[22] R. Shetty and P. Jadhav, "Predictive maintenance in industrial automation and smart manufacturing applications using artificial intelligence," International Academic Journal of Science and Engineering, vol. 10, no. 3, 2023, doi: 10.71086/IAJSE/V10I3/IAJSE1029.

[23] S. K. Shil, "AI driven predictive maintenance in petroleum and power systems using random forest regression model for reliability engineering framework," American Journal of Scholarly Research and Innovation, 2025, doi: 10.63125/477x5t65.

[24] M. Cavus, D. Dissanayake, and M. Bell, "Next generation of electric vehicles: AI-driven approaches for predictive maintenance and battery management," Energies, vol. 18, no. 5, Art. no. 1041, 2025, doi: 10.3390/en18051041.

[25] R. Guntupalli, "Optimizing cloud infrastructure performance using AI: Intelligent resource allocation and predictive maintenance," Journal of Informatics Education and Research, vol. 5, no. 2, 2023, doi: 10.52783/jier.v5i2.2939.

[26] L. Rojas, Á. Peña, and J. García, "AI-driven predictive maintenance in mining: A systematic literature review on fault detection, digital twins, and intelligent asset management," Applied Sciences, vol. 15, no. 6, Art. no. 3337, 2025, doi: 10.3390/app15063337.

[27] E. B. Nightingale, J. Elson, J. Fan, O. Hofmann, J. Howell, and Y. Suzue, "Flat datacenter storage," in *Proc. 10th USENIX Symp. Operating Systems Design and Implementation (OSDI)*, 2012, pp. 1–15.