# Using Time Series Clustering to Inform Multimodal CNN Architectures

Joshua Sylvester[1], Matthew Lee[1], Daniel Canon Ellis[2], Mitchell A. Thornton[1], and Eric C. Larson[1]
[1]Darwin Deason Institute for Cybersecurity, Southern Methodist University, Dallas, TX, USA
[2]Department of Computer Science, Southern Methodist University, Dallas, TX, USA

***Abstract*—Multimodal machine learning, in the context of deep learning, allows a neural network to process various sources of data and combine information from each data source. However, there are an exponential number of ways in which modalities can be combined for processing which can result in large architecture design searches to inform the most optimal manner of combining data streams. To mitigate this problem, we present a way to inform the creation of multimodal machine learning convolutional neural network architectures in the domain of time series datasets. Specifically, we propose the use of time series clustering as a method for informing the creation of a model's multimodal architecture. We investigate two different approaches to this method (a Euclidean- and Granger-based approach) and demonstrate effectiveness with multiple time series datasets. We find that our proposed methods can improve a model's predictive capabilities while decreasing the training time required for the model to converge. Moreover, our method eliminates the need for a costly architecture search.**

## I. Introduction

Multimodal machine learning, in the context of deep learning, describes methods for processing input modalities in distinct ways and fusing the information from each modality to best support prediction [2, 8]. This means that a model can be designed in a way that individualizes how one or more modalities are processed. Ideally, fusion methods can customize the network structure in a way that best represents the relationships within the dataset. However, identifying an optimum customized model structure is a time-consuming process, requiring all possible modality configurations to be explored (i.e., a grid or brute-force search that scales exponentially with the number of modalities). Some previous works use genetic algorithms to facilitate neural architecture searches, which are also highly computational [6, 5]. In this work, we present a method to identify relationships among modalities as a single preprocessing step that can then "inform" the creation of a multimodal model architecture. We explore this approach in the domain of time series datasets and convolutional neural network (CNN) architectures. Specifically, we propose the use of time series clustering as a method for informing the creation of a model's multimodal architecture. Using the resulting dendrogram from time series clustering, we design a CNN architecture such that modalities are fused to mirror the structure of the dendrogram. This method provides a computationally efficient alternative to brute force searches.

We first explored such a methodology in 2019 where we utilized Granger causality based time series clustering as a pre-processing step to inform the creation of wide-and-deep networks ¡todo: insert white paper ref¿. Since then we have updated the methodology to work in the context of CNNs.

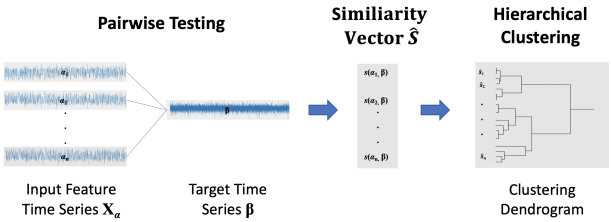


Fig. 1: The pairwise testing followed by hierarchical agglomerative clustering to produce a dendrogram that influences modality fusion.

Granger causality was designed to determine predictability between variables, specifically for determining how well one time series can forecast another [9]. Substituting this method for traditional time series clustering methods provides a novel approach for identifying meaningful relationships in time series datasets. In addition to the Granger-based clustering method, we investigate Euclidean-distance-based approach and cosine-similarity-based time series clustering approaches in a higher dimensionality dataset. For each method we construct a model architecture based on the method's resulting dendrogram, we call these the "Granger-informed", "Euclidean-informed", or "cosine-informed" models. We compare these to a "generic model" architecture (shown in Figure 2) in which the modalities are processed in an identical manner (*i.e.*, with the same number of convolutional layers containing the same kernel sizes and number of filters) before all the modalities are fused simultaneously. For all models, after modalities are fused, they are further processed through several additional convolutional layers.

Our main research question is: does a multimodal CNN architecture, informed by time series clustering, exhibit higher performance and faster training time than the generic model? Additionally, we investigate which of the clustering approaches produces a superior CNN architecture, if any. We hypothesize that the "Granger-informed" model could prove to exhibit higher performance for certain datasets. In our previous works, we have observed that Granger-Causality-based clustering can identify more complex time series relationships, even when the clustered time series are not correlated [12,

979-8-3503-5952-7/23/$31.00 ©2023 IEEE

18]. This is an artifact of the assumptions used in Granger-Causality-based clustering, where the measure of affinity is based upon common statistical influences, rather than direct distance or correlation measures.

## II. Related Work

Much of the current research into multimodal model architectures has focused on advancements in applying these architectures to different real-world datasets and analyzing how including different data modalities can affect performance. Such real-world datasets include flight, medical, spatial, text, speech or image data [14][10][1]. Other research has focused on the architectures behind multimodal models and how adjusting them can affect model performance [16]. In the paper by D. Cheng *et al.*, the authors used multi-modal architectures with a proposed multi-modality graph neural network (MAGNN) to learn from these multimodal inputs for financial time series prediction and leveraging a two-phase attention mechanism for joint optimization to increase model interpretability [4]. Research has also focused on Convolutional Neural Networks that perform feature representation learning from a concatenation of sensor types [14]. Some research has focused on aspects of the data like time series analysis that should allow a clearer insight into how the data is arranged [15].

Additionally, researchers have focused on when features should be fused in multimodal models when working with time series datasets. This work includes experimenting with several different architecture fusion methods such as late feature fusion, decision fusion, mid-fusion, fusion schemes, auxiliary supervision and temporal dropout to name a few [15]. Other work by Srivastava and Salakhutdinov utilizes a multimodal generative model, based on deep Boltzmann machines, where multimodal representations are learned via fitting the joint distributions of multimodal data over various generative models [7]. These previous works differ from our proposed work in that they require large architecture searches to find an optimal architecture configuration, whereas our proposed method discovers the architecture directly.

One work by Tan *et al.* explored using symbolic transfer entropy as an equivalent process to Granger causality to identify features to be merged prior to classification in Random Forest models [19]. We also utilize Granger causality-based clustering butour work uses the clustering procedure to identify how features should be merged within the network itself, rather than being merged prior to being input into the model. Specifically, we aim to provide a novel approach to designing mulitmodal CNN models specific to time series datasets. Using time series clustering as the method for identifying relationships within a given dataset should allow for models to be designed whose structure mirrors these identified relationships within the dataset.

## III. Methodology

Our methodology uses hierarchical agglomerative clustering (HAC) with complete linkage to cluster input time series in a dendrogram based on how they affect one or more target time series. Figure 1 demonstrates this process of pairwise testing / comparison resulting in a similarity vector which can then be used for clustering.

This dendrogram is then used to "inform" the creation of a multimodal CNN. We use the term "inform" loosely as this is a heuristic where the CNN architecture is constructed to reflect the structure of the dendrogram. The intuition is that initializing the model in this way will effectively "pre-program" relationships of interest into the network architecture. The hypothesis is that this process will help the model to distill and process information more effectively and ultimately lead to better performance than a "generic model". A "generic model" is a multimodal architecture as shown in Figure 2 in which the features are processed in the exact same manner and merged simultaneously late in the network.

We analyze three methods for performing the pairwise testing step of this process which determines the similiarity vector between the input and target time series. In the context of a low dimensional dataset we explore a Euclidean-based approach and a Granger-causality-based approach. In the context of a higher dimensional dataset we also include a third method, cosine-similarity-based approach. The three methods will be compared and contrast both with each other and with a "generic model", shown in Figure 2 and discussed in Section IIIe.

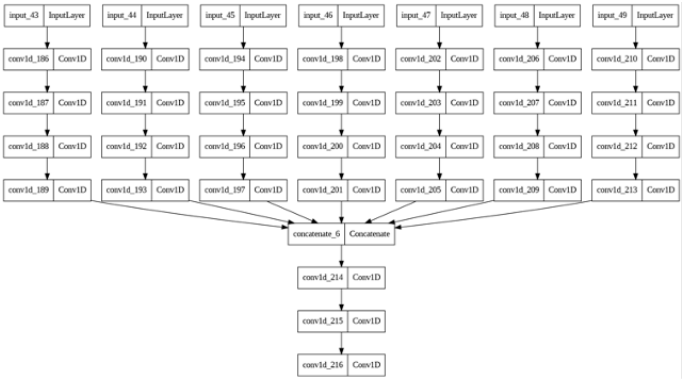


Fig. 2: Diagram of the generic model. All features are processed with the same configuration of convolutional layers. The kernal sizes and number of filters may change as a feature descends through the layers but these same layer adjustments are consistent across modalities.

.

### A. Granger-based Clustering

Granger causality is used to determine the level of forecastability that one time series has on another [9]. It does so by taking a bivariate autoregressive model consisting of variables $x$ and $y$ and testing whether the variance of the residuals increases when the values of $x$ are reduced to zero and the model becomes univariate. The formulation for Granger causality is shown in equations 1 and 2 where equation 1 represents the restricted model, or univariate case, and equation 2 represents the unrestricted model, or bivariate case.

The methodology also utilizes lags of $x$ and $y$, in equations 1 and 2, $m$ represents the maximum lag. During testing, if the variance of the residuals in the restricted model is larger than that of the unrestricted model, this suggests $x$ Granger-causes $y$ since including $x$ increased the unrestricted model's predictive capability.[9].

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + ... + a_m y_{t-m} + \epsilon_t \quad (1)$$

$$\begin{aligned} y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + ... + a_m y_{t-m} + ... \\ + a_m y_{t-m} + b_1 x_{t-1} + ... + b_m xt - m + \epsilon_t \end{aligned} \quad (2)$$

We conduct a Granger causality test between each of the input feature time series and each output time series collecting the resulting p-values [18] [12]. These p-values are then put through a logistic function transformation such that smaller p-values are mapped to larger values and larger p-values are mapped to smaller values [12]. The purpose of doing so is to make the results more intuitive where larger values indicate a higher level of Granger causality between two time series. The resulting values are then clustered using HAC to produce the dendrogram which will "inform" the Granger-informed CNN architecture.

### B. Euclidean-based Clustering

The Euclidean-based method simply measures the Euclidean distance in a pairwise manner between each input and target time series. However, instead of calculating the Euclidean distance using the entirety of an input time series and target time series, we instead take the Euclidean distance between subsections of the two time series and then average these distances together. This is done to more closely resemble how the Granger method uses subsections, or lags, of the time series. This should mean for a more direct comparison between the Granger and Euclidean methods. These resulting Euclidean distances will then be clustered using HAC to produce the dendrogram used in constructing the Euclidean-informed CNN architecture.

### C. Cosine-based Clustering

In the case of a high dimensional dataset, we also employ a cosine-similarity-based clustering technique where the cosine similarity between each input and target time series pairing is calculated. Similarly to the Euclidean-based clustering method, we also perform this process on subsections of each time series pairing and average the similarity together for each time series pairing.

### D. Datasets

We will be utilizing two datasets for our investigation. The first is an occupancy detection dataset [3] [1]. The second is an airplane maintenance prediction dataset [2]. The occupancy detection dataset exists in a lower dimensional space with 7 inputs and 1 target time series, whereas the maintenance prediction dataset exists in a higher dimensional space with 20 input time series and 20 target time series.

The occupancy dataset consists of several different environmental time series collected in an office room. These time series include light levels, temperature, humidity, and CO2. The timestamps can be used to create two additional time series containing temporal information. One is the number of seconds from midnight, denoted as NSM. The second is whether or not the current day is a weekday or weekend, denoted as week status (WS). There is a single target time series consisting of binary values indicating whether or not the office room is occupied at a given timestamp. The intuition is that these time series should exhibit very clear relationships, for example we would expect the office room to be occupied if the CO2 levels are up, there are higher light levels, and the current day is a weekday.

The occupancy dataset was published consisting of 3 sets. A training set, and two test sets. The training set has approximately 8,000 examples, the first test set has approximately 2,500 examples, and the second test set contains approximately 10,000 examples. Measurements are recorded in approximately 1 minute intervals.

The maintenance prediction dataset consists of hundreds of thousands of records of maintenance actions and error codes across several different airplanes which can be organized into time series. There are over 200 unique maintenance actions and over 1000 unique error codes. The dataset was created particularly to investigate error codes which commonly decreased following a maintenance action associated with fixing corrosion. The dataset was subset into the top-20 unique corrosion-associated maintenance action codes and the top-20 error codes which showed the most significant decrease following a corrosion-associated maintenance action. We create time series for each of the unique codes where each value in the time series represents the frequency of that code on a given day. We note that there is not an explicit prediction task for the maintenance dataset. Therefore, we use this dataset only to investigate the model size as a result of different clustering methods.

### E. Model Creation

To create an "informed" CNN architecture we utilize the dendrogram created during the corresponding clustering step to create an architecture design which mirrors the structure of the dendrogram, Figure 3 demonstrates this process. The left-side of the figure shows the dendrograms identified from the Granger and Euclidean-based clustering methods on the occupancy dataset, and the right-side shows the associated CNN model architectures. As an example of the mapping process, take the humidity and light features shown in the Granger-based clustering dendrogram. They are clustered separately

[1]The Occupancy Dataset can be found here: https://archive.ics.uci.edu/ml/datasets/Occupancy+Detection

[2]The Maintenance Prediction Dataset was accessed here in (Fall 2019) https://www.hackthemachine.ai/s/HTM_MSP_Final.csv

from the rest of the input features. This is represented in the CNN architecture such that the two feature branches are merged together initially but are not merged with the rest of the network until all other feature branches have been merged together themselves.

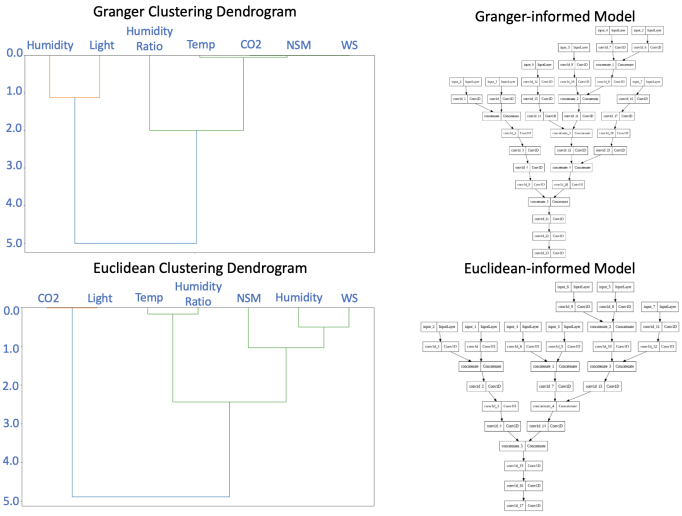


Fig. 3: Dendrograms and diagrams of the associated informed multimodal model architectures. The top row shows the Granger-based method's dendrogram and model while the Euclidean-based method's dendrogram and model is shown in the bottom row

*1) Occupancy Detection:* For the occupancy dataset experiments, each input is first put through a convolutional layer with 25 filters and a $9 \times 1$ kernel. The choice of 25 filters was made so that the input layer would have enough complexity, while also allowing room for the number of filters to increase deeper into the network without exploding. Using a kernel size of 9 allows longer temporal dependencies to be captured in the early layers. Following the input convolutions, feature branches are then merged in the order indicated by the dendrogram. After each merging of feature branches, the number of filters is increased by 25 filters to capture more complex information deeper in the network and the kernel size is reduce by 2 to focus on more specific temporal details while also balancing computational efficiency.

Additionally, when two branches consisting of $c_1$ and $c_2$ channels are merged, they are first checked to see if their respective number of filters and kernel sizes are equal, and if they are not, a convolution containing $max(c_1, c_2)$ filters is applied to the branch with fewer filters. Similarly, the minimum kernel size between the two branches is used in this additional convolution. Following this additional convolution, the two branches are concatenated. The motivation behind this branch alignment process is to provide more focus on the impact of when and where feature branch fusion occurs and ensure that differing filter counts or kernel sizes are not biasing results.

After merging all of the branches, 3 additional convolutional layers are added with the final layer acting as the output for the network (500 filters with a $5 \times 1$ kernel, 50 filters with a $3 \times 1$ kernel, and 1 filter with a $1 \times 1$ kernel). The choice of 500 filters was made to capture the large amount of features resulting from the final branch fusion in the model. The scaling down of filters and kernel size was then performed so that the output of the final convolution will represent the model output removing the need for a linear layer. Doing so, allows the model to handle variable-shaped inputs, which is of particular importance as the time series in the occupancy detection dataset are of varying lengths.

The "generic model" constructed for the occupancy dataset consists of a branch in the network for each input feature. Each of these branches consists of 4 convolutional layers which start with 25 filters and scale up to 100 filters. The kernel size starts at $7 \times 1$ and scaling down to a $1 \times 1$. Then each of the branches is concatenated and put through three additional convolutional layers, (500 filters with a $5 \times 1$ kernel, 50 filters with a $3 \times 1$ kernel, and 1 filter with a $1 \times 1$ kernel). These layer configurations were chosen to closely mirror the configuration choices made in the "informed" models.

*2) Maintenance Prediction:* For the creation of "informed" CNN architectures in the context of the maintenance prediction dataset, we perform a similar procedure to the one discussed previously with some slight modifications to the convolutional layer configurations. Each branch in the model starts with a convolution using 10 filters and a $3 \times 1$ kernel. The kernel size will stay consistent throughout the network; however, as two branches are merged, the number of filters in the next convolutional layer increases to accommodate the additional features. Due to the higher dimensionality of the dataset, we explore the use of a consistent scaling factor when scaling the number of filters following the fusion of two feature branches. We explore three different scaling factors, 1.125, 1.25, and 1.5. As an example of how this works, consider two inputs each being processed by a convolution of 10 filters and a scaling factor or 1.5. After fusing the two feature branches the next convolutional layer will contain 15 filters. As will show in Table V, the number of parameters increase rapidly with the depth dendrogams from the maintenance prediction dataset, because of this we do not apply branch alignment process discussed in the last section to reduce effects on the exploding parameter count.

### F. Method of Evaluation

For the occupancy detection dataset, we compare the "generic", Euclidean-informed, and Granger-informed models. The methods will be evaluated on the occupancy dataset and comparisons will be performed using accuracy, area under the curve (AUC), the time it takes to train the model to convergence, and the epochs it takes to train the model to convergence. The McNemar test will be used to compare the resulting models [17] [13]. The McNemar test evaluates the marginal homogeneity of two dichotomous variables, which can be used to compare two machine learning models and

evaluate whether the models' predictions are statistically different.

For the maintenance prediction dataset, we explore the Euclidean and Granger-based clustering procedures and also include a cosine-based procedure due to the higher dimensionality of the dataset. Because there was a lack of ground truth labels for this dataset, we only investigate the clustering procedure and possible model parameters. There is no explicit training task for this model.

## IV. Results

### A. Occupancy Detection Dataset Results

The Euclidean-based and Granger-based clustering methods produced two different dendrograms which in turn inform the creation of two different model architectures. These dendrograms and constructed models are shown in Figure 3.

Figure 4 shows the receiver operating characteristic (ROC) curves for the different models across the two test sets. For both test sets the models informed by time series clustering appear to have better ROC curves and AUCs than the generic model. However, the differences are very small and are likely only due to a few prediction differences. The accuracies for the various models at the optimal threshold value are reported in Table I. Also reported in Table I are results from Tan *et al.* which applied Granger Causality-based clustering on this dataset as a preprocessing step before using a Random Forest for classification. Both of the informed multimodal architectures achieve higher accuracies than the the generic model. However, both the Euclidean and Granger informed models have exactly the same accuracies on both test sets, we explore the differences in their specific predictions in Table III.

TABLE I: Model Accuracies for test sets

| Method | Test Set 1 (acc) | Test Set 2 (acc) |
|---|---|---|
| Generic Model | 97.8236% | 98.1439% |
| Euclidean-informed Model | 97.8611% | **99.4257%** |
| Granger-informed Model | 97.8611% | **99.4257%** |
| Prev SOTA [3] | 97.90% | 99.33% |
| Tan et al. [19] | 97.12% | Not Reported |

The AUCs in Figure 4 demonstrate the Granger-informed model slightly outperforms the Euclidean-informed model with AUCs 0.993802 and 0.997231 on test set 1 and 2 respectively, compared to the Euclidean-informed model's AUCs of 0.991519 and 0.996842.

TABLE II: Model Training Timing Metrics

| Model | Training Time | Epochs to conv. | Avg time per epoch |
|---|---|---|---|
| Generic Model | 73.16 min | 84 | 52.22 s |
| Euclidean-informed Model | 7.21 min | 27 | 16.03 s |
| Granger-informed Model | 12.83 min | 46 | 16.58 s |

The contingency tables shown in Table III and Table IV can be used to perform a McNemar tests to establish whether or not the differences in the models predictive capabilities

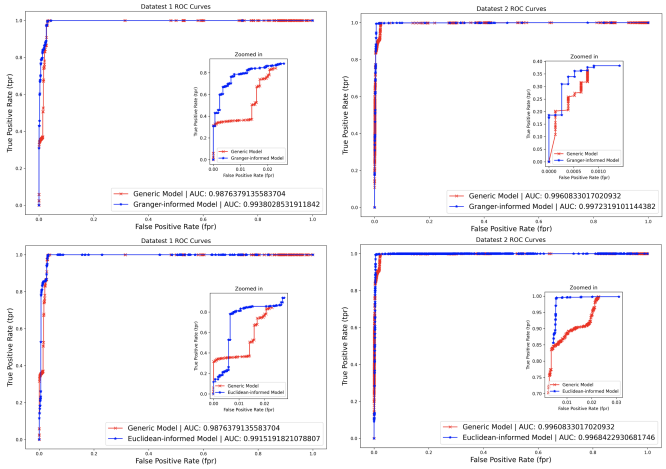


Fig. 4: ROC Curves generated during model evaluation for the different methods. The two columns represent the different test sets. The first row shows the ROC curves for the Granger-informed model vs the generic model. The second row shows the ROC Curves for the Euclidean-informed model versus the generic model. In both sets of figures, the generic model is always shown as red. AUC is noted in the legend

.

are statistically significant. In the case of comparing both of the informed models to the generic model (Table IIIa and Table IIIb), the informed models differences are statistically significant ($p < 0.01$) on the second test set and not statistically significant on the first test set ($p \approx 1$). This indicates that the proposed time series clustering-informed model will perform no worse than a generic model and may increase performance for certain datasets.

When comparing the Euclidean-informed model to the Granger-informed model (Table IIIc) we can can see that while the models have the same accuracies, they do not actually produce the same predictions. However, assessing these differences using a McNemar test yields a p-value of 1 on both test sets, indicating the differences in predictions are not statistically significant.

The contingency table shown in Table IVa compares the Granger-informed model to the current state-of-the-art method, latent Dirichlet allocation (LDA), on the occupancy detection dataset. The differences in the models' predictions are not statistically significant on the first test set, but are statistically significant for the second dataset, ($p < 0.05$). Similar results are reported in Table IVb for the Euclidean-informed model vs. LDA. The models' prediction differences are not statistically significant on the first test set but are on the second test set ($p < 0.05$).

One important takeaway is that, to achieve the results reported for the LDA approach previously mentioned, the authors performed a grid search on all possible combinations of inputs into their model. Our "informed" model creation approach bypasses the need to train a model for every feature combination because the clustering procedure identifies how

TABLE III: Contingency Tables for Occupancy Detection, Generic Model versus "informed" Models

(a) **Generic model vs. the Granger-informed model**

*Test Set 1*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| Generic Correct | 2606 | 1 |
| Generic Incorrect | 2 | 56 |

*Test Set 2*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| Generic Correct | 9564 | 7 |
| Generic Incorrect | 132 | 49 |

(b) **Generic Model vs. the Euclidean-informed model**

*Test Set 1*

| | Euc-inf Correct | Euc-inf Incorrect |
|---|---|---|
| Generic Correct | 2607 | 0 |
| Generic Incorrect | 1 | 57 |

*Test Set 2*

| | Euc-inf Correct | Euc-inf Incorrect |
|---|---|---|
| Generic Correct | 9564 | 7 |
| Generic Incorrect | 132 | 49 |

(c) **Euclidean-informed Model vs. the Granger-informed model**

*Test Set 1*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| Euc-inf Correct | 2607 | 1 |
| Euc-inf Incorrect | 1 | 56 |

*Test Set 2*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| Euc-inf Correct | 9693 | 3 |
| Euc-inf Incorrect | 3 | 53 |

TABLE IV: Contingency Tables for Occupancy Detection, SOTA versus "informed" models

(a) **SOTA LDA vs. the Granger-informed model**

*Test Set 1*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| LDA Correct | 2608 | 1 |
| LDA Incorrect | 0 | 56 |

*Test Set 2*

| | Granger-inf Correct | Granger-inf Incorrect |
|---|---|---|
| LDA Correct | 9683 | 4 |
| LDA Incorrect | 13 | 52 |

(b) **SOTA LDA vs. the Euclidean-informed model**

*Test Set 1*

| | Euc-inf Correct | Euc-inf Incorrect |
|---|---|---|
| LDA Correct | 2608 | 1 |
| LDA Incorrect | 0 | 56 |

*Test Set 2*

| | Euc-inf Correct | Euc-inf Incorrect |
|---|---|---|
| LDA Correct | 9684 | 3 |
| LDA Incorrect | 12 | 53 |

each variable should be treated with respect to one another in the network.

When considering the total training time and the numbers of epochs needed to convergence (shown in Table II), the Euclidean-informed method is significantly faster with a total training time of $7.21$ min and only $27$ epochs needed to convergence. This is near half the time and epochs required to train the Granger-informed model and is $10\times$ less training time and approximately $4\times$ fewer epochs than the generic model. While the Granger-informed model is not as efficient as the Euclidean-informed model, it is still about $5\times$ as fast as the generic model, converging in half as many epochs. These training efficiency improvements are likely due to the fact that the clustering informed models do not require near as many parameters as the generic model. The generic model contained nearly two million parameters, the Granger-informed method contained approximately $975k$ parameters, and the Euclidean-informed method approximately $800k$ parameters.

### *B. Maintenance Prediction Dataset Results*

The three clustering dendrograms are shown in Figure 5 for the Granger-based, Euclidean-based, and cosine-based clustering procedures. In this higher dimensional space, the Euclidean-based clustering clearly fails to capture any clear relationships. The cosine-based clustering does a slightly better job but still struggles for close to half of the maintenance action codes. Both the Euclidean-based and cosine-based clustering dendrograms suggest the presence of outliers. The Granger-based method seems to better handle these outliers as well as produce more robust representation of the clusters.

The maximum depth for the Euclidean-based clustering is $7$, for the cosine-based clustering is $12$, and for the Granger-based clustering is $6$. Because of the varying depths of the clusterings, creating a CNN architecture from each of these dendrograms results in three models which differ drastically in their number of parameters. Table V shows the number of parameters for each of the resulting models across a variety of scaling factors. Various scaling factors are investigated for how to alter the number of filters in each layer. A common ratio suggested by [11] is 1.125.

We can see that for each scaling factor the Granger-informed model has the fewest number of parameters. As the scaling factor grows it becomes apparent that the Euclidean-informed and cosine-informed models grow much more quickly than the Granger-informed model, further demonstrating the importance of a clustering method which is robust to outliers and can maintain a smaller tree depth.

## V. Conclusions

We show that using either a Euclidean-based or Granger-based time series clustering method to inform a multimodal CNN architecture can produce a model which has improved predictive capabilities compared to that of a generic model architecture in some cases, while not impeding its predictive capabilities in other cases. Furthermore, we show that this increase in performance of the two clustering-informed

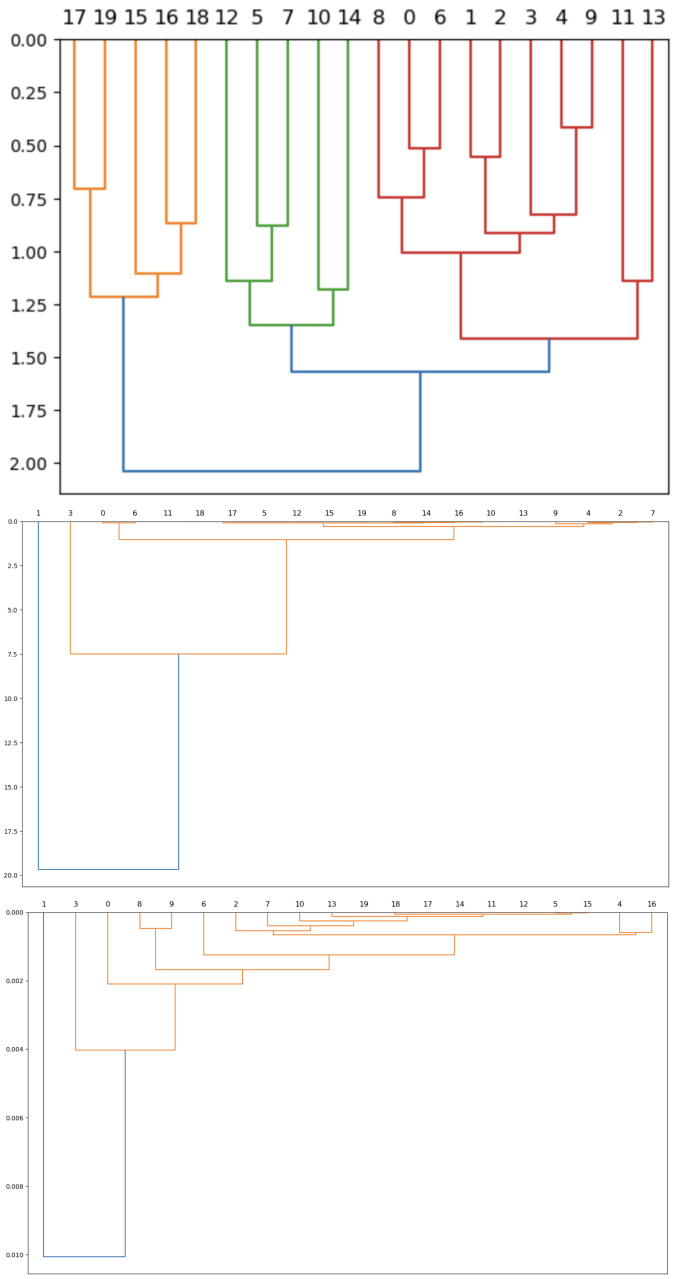


Fig. 5: Clustering dendrograms generated for maintenance prediction dataset. Granger dendrogram (top), Euclidean dendrogram (middle), and cosine dendrogram (bottom)

.

models also comes with the benefit of drastic training time improvements over that of the generic model. Contrasting the Granger-informed and Euclidean-informed models, the AUCs between the two suggest that further investigation is required to understand what tradeoffs exist for datasets with relatively few modalities, if any. On the occupancy dataset we conclude that the differences are not statistically significant. Despite this, the Euclidean-informed model requires fewer parameters to achieve the same evaluation accuracy lead to a faster

TABLE V: Maintenance prediction model parameters

| Method | Scaling Factor | # of parameters |
|---|---|---|
| Euclidean-informed Model | 1.125 | 46,159 |
| Cosine-informed Model | 1.125 | 152,962 |
| **Granger-informed Model** | **1.125** | **27,351** |
| Euclidean-informed Model | 1.25 | 193,986 |
| Cosine-informed Model | 1.25 | 2,571,192 |
| Granger-informed Model | 1.25 | 60,674 |
| Euclidean-informed Model | 1.5 | 3,522,098 |
| Cosine-informed Model | 1.5 | 446,076,676 |
| Granger-informed Model | 1.5 | 347,129 |

training time, indicating that the Euclidean-informed model has a slight advantage compared to the Granger-informed model. In higher dimensionality datasets like the maintenance datset, we show that the Granger-based clustering approach can produce informed architectures while minimizing the amount of excess parameters when compared to its Euclidean and cosine counterparts.

## REFERENCES

[1] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 140 (June 2018), pp. 20–32. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2017.11.011. URL: https://linkinghub.elsevier.com/retrieve/pii/S0924271617301818 (visited on 05/08/2023).

[2] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (Feb. 2019). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 423–443. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2018.2798607.

[3] Luis M. Candanedo and Véronique Feldheim. "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models". In: *Energy and Buildings* 112 (Jan. 15, 2016), pp. 28–39. ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2015.11.071. URL: https://www.sciencedirect.com/science/article/pii/S0378778815304357 (visited on 05/06/2023).

[4] Dawei Cheng et al. "Financial time series forecasting with multi-modality graph neural network". In: *Pattern Recognition* 121 (Jan. 2022), p. 108218. ISSN: 00313203. DOI: 10.1016/j.patcog.2021.108218. URL: https://linkinghub.elsevier.com/retrieve/pii/S003132032100399X (visited on 05/08/2023).

[5] Xinyi Ding et al. “An approach for combining multimodal fusion and neural architecture search applied to knowledge tracing”. In: *Applied Intelligence* 53.9 (2023), pp. 11092–11103.

[6] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. “Neural architecture search: A survey”. In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 1997–2017.

[7] Jing Gao et al. “A Survey on Deep Learning for Multimodal Data Fusion”. In: *Neural Computation* 32.5 (May 2020), pp. 829–864. ISSN: 0899-7667. DOI: 10.1162/neco_a_01273. eprint: https://direct.mit.edu/neco/article-pdf/32/5/829/1865303/neco_a_01273.pdf. URL: https://doi.org/10.1162/neco%5C_a%5C_01273.

[8] Jing Gao et al. “A survey on deep learning for multimodal data fusion”. In: *Neural Computation* 32.5 (2020), pp. 829–864.

[9] C. W. J. Granger. “Investigating Causal Relations by Econometric Models and Cross-spectral Methods”. In: *Econometrica* 37.3 (1969). Publisher: [Wiley, Econometric Society], pp. 424–438. ISSN: 0012-9682. DOI: 10.2307/1912791. URL: https://www.jstor.org/stable/1912791 (visited on 05/08/2023).

[10] Soo-Yeon Han et al. “Classification of pilots’ mental states using a multimodal deep learning network”. In: *Biocybernetics and Biomedical Engineering* 40.1 (Jan. 1, 2020), pp. 324–336. ISSN: 0208-5216. DOI: 10.1016/j.bbe.2019.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0208521619304887 (visited on 05/08/2023).

[11] Forrest N Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[12] Matthew Lee et al. “Side Channel Identification using Granger Time Series Clustering with Applications to Control Systems”. In: *Proceedings of the 8th International Conference on Information Systems Security and Privacy, ICISSP 2022, Online Streaming, February 9-11, 2022*. Ed. by Paolo Mori, Gabriele Lenzini, and Steven Furnell. SCITEPRESS, 2022, pp. 290–298. DOI: 10.5220/0010781600003120. URL: https://doi.org/10.5220/0010781600003120.

[13] Quinn McNemar. “Note on the sampling error of the difference between correlated proportions or percentages”. In: *Psychometrika* 12.2 (June 1947), pp. 153–157. DOI: 10.1007/bf02295996. URL: https://doi.org/10.1007/bf02295996.

[14] Valentin Radu et al. “Multimodal Deep Learning for Activity and Context Recognition”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (Jan. 8, 2018), pp. 1–27. ISSN: 2474-9567. DOI: 10.1145/3161174. URL: https://dl.acm.org/doi/10.1145/3161174 (visited on 05/08/2023).

[15] Vivien Sainte Fare Garnot, Loic Landrieu, and Nesrine Chehata. “Multi-modal temporal attention models for crop mapping from satellite time series”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 187 (May 2022), pp. 294–305. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2022.03.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0924271622000855 (visited on 05/08/2023).

[16] Shaker El-Sappagh et al. “Multimodal multitask deep learning model for Alzheimer’s disease progression detection based on time series data”. In: *Neurocomputing* 412 (Oct. 2020), pp. 197–215. ISSN: 09252312. DOI: 10.1016/j.neucom.2020.05.087. URL: https://linkinghub.elsevier.com/retrieve/pii/S0925231220309383 (visited on 05/08/2023).

[17] Joshua Henrina Sundjaja, Rijen Shrestha, and Kewal Krishan. “McNemar And Mann-Whitney U Tests”. In: *StatPearls*. Treasure Island (FL): StatPearls Publishing, 2023. URL: http://www.ncbi.nlm.nih.gov/books/NBK560699/ (visited on 07/08/2023).

[18] Joshua Sylvester et al. “Time Series Clustering Using Granger Causality to Identify Time Series Applicable for Forecasting Internal Waves in Lake and Marine Environments”. In: 2021 (Dec. 1, 2021). Conference Name: AGU Fall Meeting Abstracts ADS Bibcode: 2021AGUFMIN45A..09S, IN45A–09. URL: https://ui.adsabs.harvard.edu/abs/2021AGUFMIN45A..09S (visited on 05/08/2023).

[19] Sin Yong Tan et al. “Granger Causality Based Hierarchical Time Series Clustering for State Estimation∗∗This work was authored in part by Anthony R. Florita from National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AR0000938. Funding provided by U.S. Department of Energy, Advanced Research Projects Agency-Energy (ARPA-E). The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government.” In: *IFAC-PapersOnLine*. 21st IFAC World Congress 53.2 (Jan. 1, 2020), pp. 524–529. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2020.12.324. URL: https://www.sciencedirect.com/science/article/pii/S2405896320306054 (visited on 11/16/2023).