

DEEP LEARNING APPROACHES FOR BSM PHYSICS: EVALUATING DNN AND GNN PERFORMANCE IN PARTICLE COLLISION EVENT CLASSIFICATION*

ALI CELIK

Department of Physics, Burdur Mehmet Akif Ersoy University, Burdur, Turkey

*Received 19 August 2024, accepted 7 November 2024,
published online 21 November 2024*

Detecting beyond the Standard Model (BSM) signals in high-energy particle collisions presents significant challenges due to complex data and the need to differentiate rare signal events from the Standard Model (SM) backgrounds. This study investigates the efficacy of deep learning models, specifically Deep Neural Networks (DNNs) and Graph Neural Networks (GNNs), in classifying particle collision events as either BSM signal or background. The research utilized a dataset comprising 214,000 SM background and 10,755 BSM events. To address class imbalance, an undersampling method was employed, resulting in balanced classes. Three models were developed and compared: a DNN and two GNN variants with different graph construction methods. All models demonstrated high performance, achieving Area Under the Receiver Operating Characteristic curve (AUC) values exceeding 94%. While the DNN model slightly outperformed GNNs across various metrics, both GNN approaches showed comparable results despite different graph structures. The GNNs' ability to explicitly capture inter-particle relationships within events highlights their potential for BSM signal detection.

DOI:10.5506/APhysPolB.55.10-A2

1. Introduction

The Standard Model of particle physics [1, 2] has been extensively tested by the CMS and ATLAS experiments, with experimental data confirming its remarkable success. However, despite its achievements, the SM cannot explain some phenomena, which leads us to search for beyond the Standard Model theories. At CERN, the CMS and ATLAS collaborations have conducted searches for dark matter candidates, one of the primary phenomena that the SM cannot explain. While these searches have not yielded positive results thus far, they have placed mass limits on the sought-after particles. In

* Funded by SCOAP³ under Creative Commons License, CC-BY 4.0.

one of these studies [3], in the scenario where a pair of gluinos is produced and these gluinos decay to the LSP via off-shell stop or sbottom, gluino masses below 2.44 TeV and 2.35 TeV, respectively, have been excluded at 95% C.L. for the case of a massless lightest supersymmetric particle called LSP ($\tilde{\chi}_1^0$).

Although the production cross section of colored particles in hadron colliders is higher than that of electroweakinos, the limits placed on these particles are also higher compared to electroweakinos. However, the mass limits placed on electroweakinos are still within the reach of CMS or ATLAS experiments. In a study [4] examining the direct production mechanism of chargino–neutralino ($\tilde{\chi}_1^\pm/\tilde{\chi}_2^0$), a mass limit of 820 GeV was placed on the ($\tilde{\chi}_1^\pm/\tilde{\chi}_2^0$) for the case of a massless $\tilde{\chi}_1^0$. A separate study [5], which investigated slepton (\tilde{l}) pair production, where each slepton decays into a lepton and an LSP, excluded slepton masses up to 700 GeV.

In another study [6] conducted by the ATLAS Collaboration, which investigated SUSY scenarios with R -parity violation, a comprehensive analysis was performed using 139 fb^{-1} of data. The analysis examined various production mechanisms, including gluino pair production, stop pair production, chargino–neutralino production, and neutralino pair production. The findings placed significant constraints on SUSY particle masses. For scenarios with high LSP mass, gluino masses up to 2.4 TeV were excluded. In cases with low LSP mass, the exclusion limit for gluino masses reached up to 2 TeV. Notably, the study also set stringent limits on the top squark mass, excluding it up to 1.35 TeV, further constraining the parameter space for supersymmetric models. Furthermore, the study provided important insights into the mass limits of the LSP itself. For higgsino LSP scenarios, where the LSP is assumed to be mass-degenerate with the second-lightest neutralino, LSP masses up to 320 GeV were excluded. In the case of wino LSP, where the LSP is assumed to be mass-degenerate with the lightest chargino, the exclusion limit for LSP masses was extended up to 365 GeV.

Given the challenges in detecting electroweakinos due to their lower production rates, researchers have turned to advanced analytical techniques to enhance their search capabilities. Therefore, different research groups need to utilize various methods to increase acceptance in studies addressing this production mechanism. One of these methods is to utilize machine learning algorithms. Machine learning algorithms can be successful in capturing linear or non-linear relationships between features in scientific fields like particle physics, where the collected data is multi-dimensional and the classical cut-and-count method may not yield sufficiently good results. A study conducted in [7] reproduced the results of studies [5, 8] performed by the ATLAS group, using machine learning algorithms, which addressed direct slepton production or chargino/neutralino production and their decay into

different final states via \tilde{l}/W^\pm boson. The obtained results showed that the machine learning method enhanced the outcomes. Machine learning algorithms are not only used in physics analyses but are also widely utilized in the reconstruction of physics objects and in the jet-flavor tagging stage in the CMS and ATLAS experiments [9–11] as well.

Machine learning algorithms have significantly contributed to the literature in many studies aimed at separating various BSM signals from backgrounds, not only using CMS or ATLAS data but also using simulated data. In the study [12], researchers developed models to separate signals from backgrounds using various machine learning algorithms, neural networks, and deep learning algorithms for two different beyond the Standard Model signals. One of these signals is the theoretical heavy and neutral Higgs boson production through gluon–gluon fusion, and the other is chargino pair production where these charginos subsequently decay into an LSP, lepton, and neutrino via a W boson. In addition to this study, researchers in [13–15] developed models using CNN, deep learning, transfer learning, and machine learning algorithms to separate the Standard Model background from BSM signals or to isolate 2D histograms representing only the SM background from 2D histograms where the signal is embedded in the SM background. These studies not only compare the performances of the developed models against each other but also demonstrate the successful application of transfer learning in cases with limited data, particularly as shown in [14].

Recent studies have highlighted the effectiveness of geometric deep learning techniques, especially Graph Neural Networks (GNNs), in tackling various track reconstruction challenges in high-energy particle physics [16–18]. Researchers in work [18] have applied a physics-inspired Interaction Network (IN), a specific type of GNN, to address particle tracking under the intense pileup conditions anticipated at the high-luminosity Large Hadron Collider (HL-LHC). By modeling particle tracking data as graphs, where silicon tracker hits serve as nodes and particle trajectories as edges, the IN GNN demonstrated remarkable edge-classification accuracy and tracking efficiency. Notably, the compact IN architecture, significantly smaller than previous GNN tracking models, shows promise for implementation in environments with limited computational resources. The study also investigated the acceleration of IN implementations using diverse computing resources, including FPGAs, and explored alternative GNN approaches that eliminate the need for predefined graph structures. These results emphasize the potential of GNNs, particularly the IN model, as a powerful tool for particle tracking and other reconstruction tasks at the HL-LHC, positioning them as strong candidates for future high-energy physics experiments

The multi-dimensional nature of particle collision data, coupled with the rarity of BSM events, necessitates advanced techniques capable of extracting subtle patterns and features. GNNs offer a promising approach due to their ability to explicitly capture the relationships between particles in each event. This study aims at leveraging the strengths of GNNs in handling complex, graph-structured data to improve the sensitivity and efficiency of BSM signal detection. By comparing the performance of GNNs with traditional deep learning methods, this research seeks to provide valuable insights into the most effective approaches for analyzing particle collision data. Hence, in this study, the classification of gluino–gluino production, a beyond-Standard-Model signal, from certain backgrounds of the Standard Model in particle physics, using graphical neural networks [19], is addressed. Although GNNs have a wide range of applications, namely in solving many graph-related machine learning problems such as node classification, link prediction, anomaly detection, and graph classification, this study focuses on the latter constructed from tabular data belonging to each class. The performances of the GNNs are then compared with each other and with that of a deep learning method.

2. Methodology

The search for BSM signals in particle physics has traditionally relied on conventional methods such as cut-and-count or shape analysis, employed by the researchers. However, the increasing complexity and volume of data generated by modern particle colliders present new challenges that demand more sophisticated analytical approaches. The multi-dimensional nature of particle collision data, coupled with the rarity of BSM events, necessitates advanced techniques capable of extracting subtle patterns and features. In this context, machine learning algorithms, particularly deep learning models, have emerged as powerful tools for analyzing complex datasets in particle physics. This study explores the application of both DNN and GNNs to the challenge of BSM signal detection. By leveraging these advanced machine learning techniques, the aim is to enhance the sensitivity and efficiency of BSM searches, potentially uncovering signals that might elude traditional analysis methods. The comparison between DNNs and GNNs also provides valuable insights into the most effective approaches for handling the unique characteristics of particle collision data, paving the way for more robust and adaptable analysis techniques in high-energy physics.

2.1. Proposed method

This study employs both DNN and GNN models to classify particle collision events, focusing specifically on BSM signal detection against back-

ground events. The analysis utilizes background and the BSM signal data provided by the research group [20, 21]. The study focused on Channel 1, where the final state consists of Missing Transverse Energy (MET), and jets. The dataset itself includes a bunch of different BSM processes such as SUSY and non-SUSY. However, only SUSY gluino–gluino pair production, which eventually decays to jets and neutralinos is considered in this work. Gluinos are assumed to have a mass of 1.6 TeV and the neutralinos have a mass of 800 GeV. The Channel 1, looking for hadronic activity, includes 214,000 SM events, and a signal sample of 10,755 is constructed as follows:

- $E_T^{\text{miss}} \geq 200 \text{ GeV}$,
- $H_T \geq 600 \text{ GeV}$,
- $E_T^{\text{miss}}/H_T \geq 0.2$,
- $N(j_{p_T > 50 \text{ GeV}}) \geq 4$,
- $N(j_{p_T > 200 \text{ GeV}}) \geq 1$,

where j can either be a jet or a b -jet. Comprehensive details regarding the dataset’s production and the pre-selection cuts applied can be found in referenced article [21].

The model inputs were defined using key physical features of particle collision events: energy (E), transverse momentum (p_T), pseudorapidity (η), and azimuthal angle (ϕ) for each physics object of interest, including jets, b -jets, electrons, and muons. Additionally, global event features such as MET and its azimuthal angle (MET_ϕ) were included. Each event was labeled with a binary target variable (1–0) indicating whether it was a BSM signal (1) or background events (0). However, particle collision events typically contain varying numbers of particles, resulting in inconsistent feature sets across events. To address this variability and ensure consistent input for both the DNN and GNN models, a pre-processing step was implemented. The input features were standardized to include a maximum of four b -jets and four jets per event. For events with more than four objects of a given type, only the first four were retained, while the total count of each object type (*e.g.*, n_{bjets} , n_{jets}) was recorded as an additional feature. For events with fewer than four objects, a feature padding technique was employed by adding placeholder values of -9999 for any missing particle data, ensuring a consistent set of features for each event. This approach balances the need for detailed event information with the requirement for uniform input across all events. Both the DNN and GNN models utilize the same pre-processed dataset, allowing for a fair comparison of their performance.

To address class imbalance, undersampling was performed using the `pandas` library [22], equalizing the number of signal and background events. For binary classification purposes, all backgrounds contributing to the signal region were treated as a single background class. The dataset was then split into training, validation, and test sets in a ratio of 60%, 20%, and 20%, respectively. Subsequent to the splitting, the training set was normalized using the ‘StandardScaler’ class from the `scikit-learn` library [23], ensuring a mean of 0 and a standard deviation of 1 for all features. The same scaling parameters computed from the training set were then applied to normalize the validation and test sets. This normalization step is crucial in preventing bias in machine learning algorithms.

A DNN model with a flexible architecture was created using Keras API [24]. The model consists of an input layer, four hidden dense layers with units of 128, 32, 64, and 192, respectively, each employing the ReLU activation function, a dropout layer with a rate of 0.2 applied after the third dense layer (64 units) to mitigate overfitting by randomly setting 20% of the input units to zero during training, and an output layer with a sigmoid activation function for binary classification (see figure 1 for the architecture of the DNN model). Keras Tuner [25] was used for hyperparameter optimization, searching for the best combination of number of layers, units per layer, dropout rates, and learning rate. The model is compiled using the Adam optimizer with a learning rate of approximately 0.00043 and the binary crossentropy loss function. Training is conducted for a maximum of

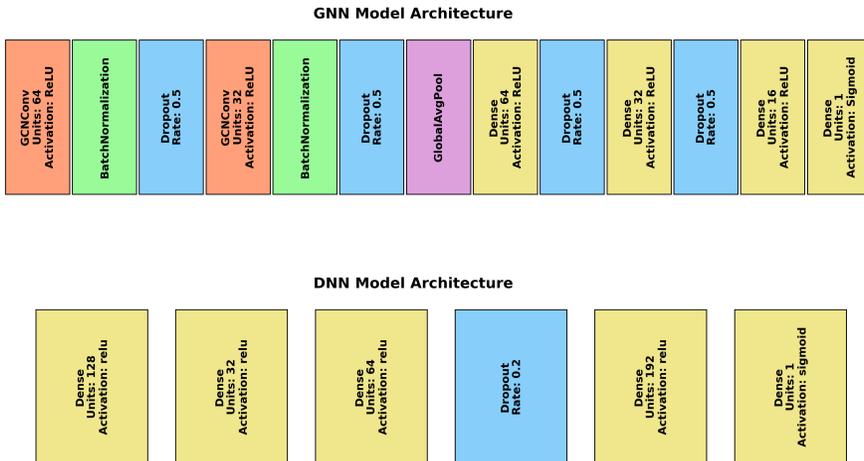


Fig. 1. Comparison of the architectures of the GNN and DNN models. The GNN model features GCN, Batch Normalization, dropout, Global Average Pooling, and dense layers, while the DNN model includes dense and dropout layers.

100 epochs, with early stopping implemented to halt training when the validation loss ceases to improve, using patience of 20 epochs. The training was conducted with a batch size set to 32. To prevent overfitting, early stopping is employed, and when triggered, it ensures that the best model is saved. Hyperparameters that are used and optimized for the final DNN model are listed in Table 1. The final model was created using the best hyperparameters and trained on the training set, validated on the validation set. The model’s performance was evaluated on the test set to ensure generalizability. A threshold of 0.5 was applied to the sigmoid output to determine class assignments. Outputs exceeding this threshold were classified as the positive class (1), while those below were assigned to the negative class (0). This consistent threshold was used across all performance metric calculations, including accuracy, precision, recall, and F1 score. The comprehensive training strategy and hyperparameter optimization resulted in a robust DNN model capable of accurately classifying particle collision events based on the provided features.

Table 1. Optimum Hyperparameters for GNN and DNN models.

Hyperparameter	GNN	DNN
Optimizer	Adam	Adam
Activation function	ReLU/Sigmoid	ReLU/Sigmoid
Loss function	Binary crossentropy	Binary crossentropy
Batch size	128	32
Learning rate	0.001	0.00043
Early stopping	yes	yes
Number of units	64–32–64–32–16–1	128–32–64–192–1
Number of layers	13	6

As part of the methodology, two unique graph structures were developed for each event in the dataset. The first structure, named GNN-1, featured MET as the central node, connected to all other feature nodes. The second structure, called GNN-2, was a hybrid connection structure. Most nodes, including MET, are fully interconnected with each other. However, certain nodes maintain specific, physically meaningful connections: the number of electron/positron (n_{e^\mp}) node links only to the electron/positron (e^\mp) node; the number of muon/antimuon (n_{μ^\mp}) node links only to the muon/antimuon (μ^\mp) node; the number of jets (n_{jets}) node connects only to jet nodes (j_1, j_2, j_3, j_4); and the number of b -jets ($n_{b\text{jets}}$) node links only to b -jet nodes (b_1, b_2, b_3, b_4). These graph structures were created using the NetworkX

library [26], with node features converted to NumPy arrays [27]. The Spektral library [28], which is based on the Keras API and TensorFlow 2, was then used to adapt these graph structures and features for use in the GNN models. The illustrations of the graphs for both cases are shown in figures 2 and 3, respectively.

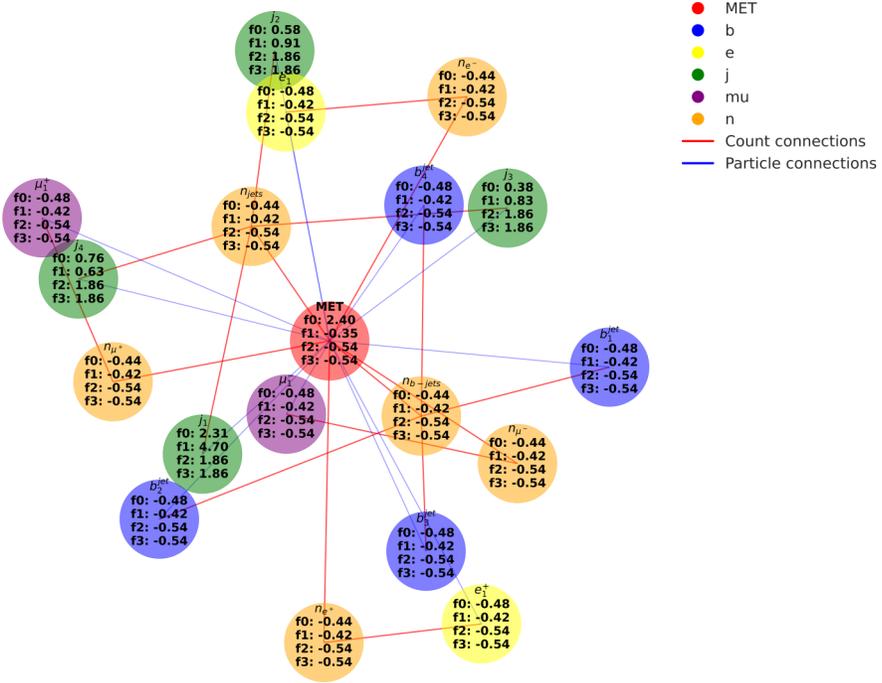


Fig. 2. Graph representation of the GNN-1 model. The Missing Transverse Energy (MET) node is centrally positioned and connected to all other nodes, forming a star-like topology. This structure emphasizes the global influence of MET on all other particles in the event.

To effectively manage and prepare the data for training and validation, a custom data loader was implemented. This data loader organized the graph structures into the training, validation, and test sets, ensuring that the data was correctly formatted and optimized for the GNN architecture. By handling the batching and shuffling of the data, the custom loader facilitated efficient training processes and consistent evaluation across the models.

Building on the developed graph structures, two GNN models were implemented to leverage these configurations. The GNN architectures comprise a total of six trainable layers with a sequence of units as 64–32–64–32–16–1. The models, created using Keras and Spektral libraries, consist of input

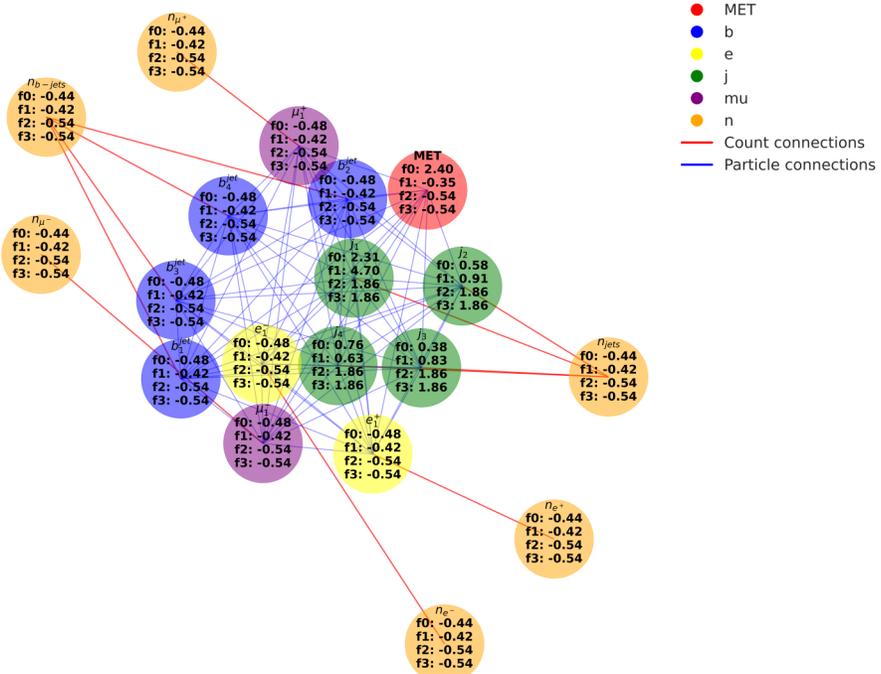


Fig. 3. Graph representation of the GNN-2 model. The graph features a hybrid connection structure. Most nodes, including MET, are fully interconnected with each other. However, certain nodes maintain specific, physically meaningful connections: the number of electron/positron (n_{e^\mp}) node links only to the electron/positron (e^\mp) node; the number of muon/antimuon (n_{μ^\mp}) node links only to the muon/antimuon (μ^\mp) node; the number of jets (n_{jets}) node connects only to jet nodes (j_1, j_2, j_3, j_4); and the number of b -jets (n_{bjets}) node links only to b -jet nodes (b_1, b_2, b_3, b_4).

layers, followed by two Graph Convolutional Network layers with 64 and 32 units, respectively, each followed by a batch normalization layer and a dropout layer with a rate of 0.5 to prevent overfitting. After the GCN layers, a global average pooling layer is applied to aggregate the node features. The model then includes three dense layers with 64, 32, and 16 units employing the ReLU activation function. Each of the first two dense layers is followed by a dropout layer with a rate of 0.5. The final output layer is a dense layer with 1 unit and a sigmoid activation function, suitable for binary classification tasks. The architecture of the model is depicted in figure 1.

Unlike the DNN model, the GNN models were optimized through manual trial and error rather than using tools like Keras Tuner. This approach involved iteratively adjusting the model architecture and hyperparameters based on validation performance. The model was compiled using the Adam optimization algorithm with a learning rate of 0.001 and trained with binary cross-entropy loss. The training was conducted for a maximum of 500 epochs, with early stopping implemented to halt training when the validation loss ceased to improve after 20 consecutive iterations. When early stopping was triggered during the training, the best model was saved. The performance of the best model was evaluated on the test set. As with the DNN model, a 0.5 threshold was applied to the GNN outputs for class determination and performance metric calculations, ensuring consistency in evaluation across all models.

The performance of the two different graph structure types was compared, and the results are presented in Table 1. Through meticulous architectural design and manual optimization, the GNN models demonstrated strong performance in classifying particle collision events, highlighting the effectiveness of graph-based approaches in distinguishing BSM signals from background events.

2.2. GNN

A Graph Neural Network is a type of deep learning model designed to work with graph-structured data. Unlike traditional neural networks that operate on grids or sequences, GNNs are capable of handling complex relationships between nodes and edges in a graph. This makes them particularly useful in domains where data is naturally represented as graphs, such as social networks, molecular structures, traffic networks, and recommendation systems. GNNs have been shown to be effective in tasks such as node classification, graph classification, and link prediction.

GNNs are widely used in various applications, including computer vision, natural language processing, and recommender systems [29–32]. For instance, in computer vision, GNNs can analyze the relationships between objects in an image, while in natural language processing, they can model the relationships between words in a sentence. In recommender systems, GNNs can be used to model the relationships between users and items, and generate personalized recommendations. Additionally, GNNs have been applied to various scientific domains, such as chemistry, biology, and physics, to analyze complex systems and make predictions [33–36].

Graph Convolutional Networks are a specific class of neural networks designed to operate on graph-structured data. GCNs can capture the complex relationships and dependencies between nodes in a graph. Mathematically, a GCN extends the concept of convolution from Euclidean space to non-

Euclidean space. Given a graph $G = (V, E)$ with V set of nodes and E as the set of edges, the graph can be represented by an adjacency matrix A and a feature matrix X . The key operation in a GCN is the graph convolution, which can be expressed as

$$H^{(l+1)} = \sigma \left(\hat{A}H^{(l)}W^{(l)} \right) .$$

Here, $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ is the normalized adjacency matrix, \tilde{D} is the degree matrix of \tilde{A} , $H^{(l)}$ denotes the feature matrix at layer l , $W^{(l)}$ is the trainable weight matrix at layer l , and σ is a non-linear activation function such as ReLU. This operation effectively aggregates the features of neighboring nodes, allowing the network to learn representations that capture the local graph structure.

2.3. DNN

Deep Neural Networks represent a significant advancement in machine learning, offering a sophisticated approach to modeling complex systems. Inspired by the neural architecture of the human brain, DNNs extend this concept with multiple hidden layers, enabling them to excel at learning intricate patterns in data.

The architecture of a DNN is characterized by its layered structure, where each successive layer processes and refines the output from the previous one. This hierarchical processing allows DNNs to extract increasingly abstract features or representations from the input data, facilitating the modeling of non-linear relationships that often challenge traditional algorithms.

The versatility of DNNs has led to their widespread adoption across various domains. In computer vision, they have revolutionized tasks such as object recognition, image classification, and even image synthesis. Similarly, the field of natural language processing has seen remarkable progress through the application of DNNs in areas such as machine translation, sentiment analysis, and conversational AI.

Moreover, the impact of DNNs extends beyond these computational fields. In healthcare, they are instrumental in advancing diagnostic techniques and accelerating drug discovery processes. The financial sector leverages DNNs for fraud detection and risk assessment, demonstrating their utility in handling complex, multidimensional data.

The key strength of DNNs lies in their ability to distill meaningful insights from large, complex datasets. This capability positions them as invaluable tools in our increasingly data-driven world, where the extraction of actionable information from vast amounts of data is crucial for advancement in numerous fields of study and industry.

3. Findings

In this study, a total of three models, two of which are GNNs, were developed, and their performances are presented comparatively. In both GNN-1 and GNN-2 models, each simulated event corresponds to a graph, with each particle or, in other words, a physics object corresponding to a node, and attributes related to these objects are referred to as features. Since missing transverse momentum is an important parameter in BSM research, in the first GNN model created, graphs were constructed with the MET node at the center and connected to other nodes, while in the second type, a model was created by selecting all nodes to be interconnected. Looking at the results of both of these models, it is clear that they provide similar performance across all the metrics.

Examining the loss and accuracy plots in figure 4, both training and validation loss follow similar trends for all three models, and training was terminated with early stopping as there was no improvement in validation loss after a certain epoch. It is also worth noting that, based on these loss graphs, overfitting does not appear to be an issue for any of the models.

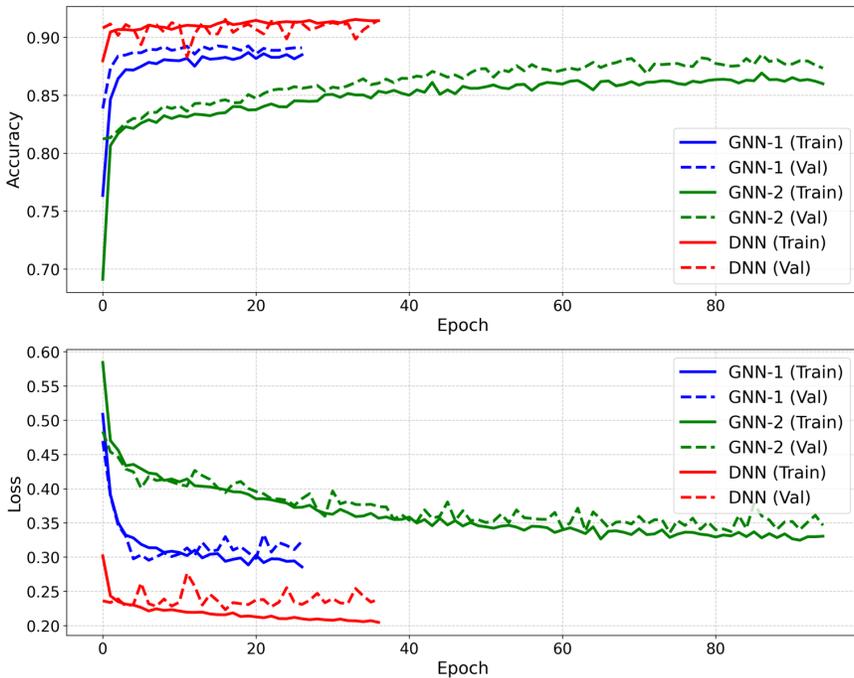


Fig. 4. Comparison of model performance: Top — accuracy of GNN-1, GNN-2, and DNN models on training and validation data; Bottom — training and validation loss over epochs for all three models.

Additionally, as shown in Table 2, the DNN demonstrates slightly better results in terms of both accuracy and other measures. However, no significant difference was observed when compared to the GNN models.

Table 2. Comparison of performance metrics for GNNs and DNN models. The metrics include accuracy, recall, precision, F1 score, and area under the curve (AUC).

Model	Accuracy	Precision	Recall	F1 Score	AUC
GNN-1	88.7%	88.4%	89.2%	88.8%	95.1%
GNN-2	87.7%	88.2%	87.1%	87.7%	94.3%
DNN	91.0%	91.2%	90.7%	91.0%	96.7%

In addition to recall, precision, accuracy, F1, the AUC was also used as a performance metric in this study. AUC is a metric that takes values between 0.5 and 1.0 and is defined as the area under the true positive rate (TPR) *versus* false positive rate (FPR) graph. While a value of 0.5 indicates performance no better than random guessing, an AUC value of 1.0 suggests that the model is an excellent classifier. A value of 0.7 can be interpreted as the model being successful. As shown in Table 2, all three models, DNN, GNN-1, and GNN-2, demonstrated excellent performance, with AUC values exceeding 94%, with the DNN achieving slightly better performance.

To further evaluate the models' performance, confusion matrices were generated for each of the three models (see Table 3). These matrices provide a detailed breakdown of true positives, true negatives, false positives, and false negatives, offering insight into the specific strengths and weaknesses of each model in classifying signal and background events.

Table 3. Confusion Matrices for GNN-1, GNN-2, and DNN models in classifying background and signal events.

GNN-1	Actual	Predicted			GNN-2	Actual	Predicted				
			BG's	Signal			Total		BG's	Signal	Total
		BG's	1896	252			2148	BG's	1896	252	2148
		Signal	233	1921			2154	Signal	279	1875	2154
	Total:	2129	2173	4302		Total:	2175	2127	4302		
DNN	Actual	Predicted									
			BG's	Signal	Total						
		BG's	1960	188	2148						
		Signal	200	1954	2154						
	Total:	2160	2142	4302							

Figure 5 provides further insight into the models' performance by plotting the background rejection as a function of signal efficiency. This representation, often referred to as a ROC curve in particle physics contexts, allows for a more nuanced comparison of the models' discriminative power across different operating points. Notably, all three models exhibit strong performance. The DNN model shows slightly better performance, with its curve slightly positioned above those of the GNN variants across most of the signal efficiency range. This is consistent with its higher AUC value. The similarity in performance between the DNN and GNN models suggests that both approaches are capable of effectively capturing the complex relationships in particle collision data. The strong performance of the GNN models indicates that the graph-based representation of collision events preserves important topological information, which can be effectively leveraged for signal-background discrimination. Notably, the models achieved high classification accuracy using only low-level features, without relying on high-level engineered variables. This finding suggests that even without high-level features, the models can effectively capture the necessary information for

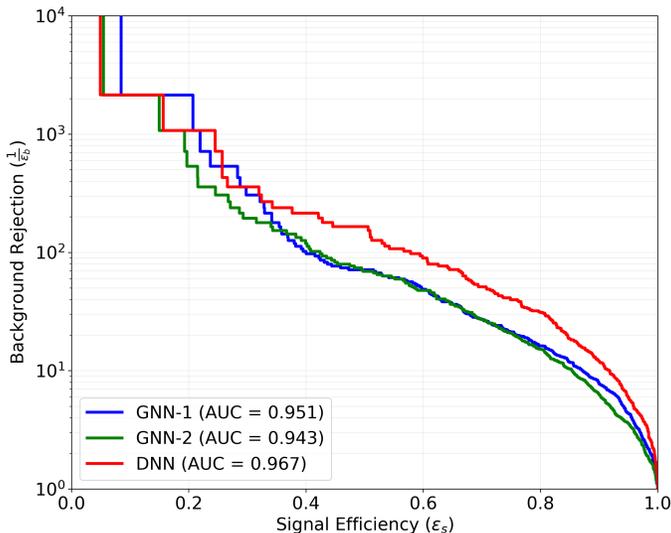


Fig. 5. Comparison of model performance: ROC curves showing signal efficiency *versus* background rejection for two GNN's (GNN-1, GNN-2) and DNN. DNN shows the highest area under the curve (AUC) of 0.967, outperforming both GNN models, particularly beyond the signal efficiency of 0.4, where it achieves a slightly higher background rejection. GNN-1 (AUC = 0.951) slightly outperforms GNN-2 (AUC = 0.943). All models converge at very high-signal efficiencies and show similar trends at low-signal efficiency regions, indicating comparable performance in both high-recall and high-precision scenarios.

accurate classification. This represents a significant advantage in simplifying data preparation and potentially reducing the need for extensive feature engineering and analysis.

Beyond the statistical performance, these results have important physics implications. The ability of both DNN and GNN models to accurately distinguish between SM and BSM events suggests that such machine learning techniques can enhance the sensitivity of BSM searches. By effectively identifying rare signal events, these models could contribute to refining current experimental limits on BSM theories, potentially leading to the discovery of new physics phenomena.

4. Conclusion

This study has demonstrated the efficacy of both DNN and GNNs in classifying particle collision events, with a focus on detecting beyond the Standard Model signals against background events. The comparative analysis of these models offers valuable insights into their performance and potential applications in high-energy physics research.

The strong performance of the GNN models, despite different graph construction methods, highlights the potential of graph-based representations in preserving crucial topological information about particle collisions. The application of GNNs presents a novel approach to particle physics analyses, reflecting the data structure more accurately by modeling the interactions between particles as graphs. This allows GNNs to capture complex dependencies and relational information inherent in collision events, which are often challenging to address with traditional methods.

By leveraging the inherent connections in the data, GNNs offer a unique advantage in detecting subtle patterns and correlations that may be indicative of new physics phenomena. This suggests that GNNs could be a valuable tool in future particle physics analyses, offering a complementary approach to traditional methods. It is worth noting that while the DNN model provided slightly better results in this study, the performance of GNNs could potentially be enhanced with access to larger datasets and further optimization, underscoring their promise in this field for advancing BSM searches.

The study's methodology, including undersampling to address class imbalance and careful feature engineering, proved to be effective in creating models that generalize well to unseen data. The high recall and precision scores across all models indicate their ability to accurately identify rare signal events, a crucial capability in BSM physics research. While this study focused on gluino–gluino pair production, the strong performance of these models suggests potential applicability to other BSM processes. This versatility, coupled with the rapid advancements in machine learning, positions

these techniques at the forefront of high-energy physics research. As the field continues to evolve, deep learning approaches like those explored in this study are poised to play an increasingly important role in pushing the boundaries of our understanding.

Building on these promising results, future research should explore the scalability and adaptability of these models. Particularly, incorporating GNNs into larger, more representative datasets typical in full-scale particle physics experiments could further enhance their effectiveness. Such advancements may contribute significantly to refining the current limits of BSM searches and potentially uncovering new physics. This study underscores the potential of deep learning approaches in particle physics analysis and motivates further investigation of GNN architectures for BSM signal detection.

REFERENCES

- [1] A. Salam, «Weak and electromagnetic interactions», *Conf. Proc. C* **680519**, 367 (1968).
- [2] S. Weinberg, «A model of leptons», *Phys. Rev. Lett.* **19**, 1264 (1967).
- [3] ATLAS Collaboration (G. Aad *et al.*), «Search for supersymmetry in final states with missing transverse momentum and three or more b -jets in 139 fb⁻¹ of proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector», *Eur. Phys. J. C* **83**, 561 (2023), [arXiv:2211.08028 \[hep-ex\]](#).
- [4] ATLAS Collaboration (G. Aad *et al.*), «Searches for new phenomena in events with two leptons, jets, and missing transverse momentum in 139 fb⁻¹ of $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector», *Eur. Phys. J. C* **83**, 515 (2023).
- [5] ATLAS Collaboration (G. Aad *et al.*), «Search for electroweak production of charginos and sleptons decaying into final states with two leptons and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions using the ATLAS detector», *Eur. Phys. J. C* **80**, 123 (2020).
- [6] ATLAS Collaboration (G. Aad *et al.*), «Search for R-parity-violating supersymmetry in a final state containing leptons and many jets with the ATLAS experiment using $\sqrt{s} = 13$ TeV proton–proton collision data», *Eur. Phys. J. C* **81**, 1023 (2021).
- [7] M. Anderssen, «Performance of Deep Learning in Searches for New Physics Phenomena in Events with Leptons and Missing Transverse Energy with the ATLAS Detector at the LHC», Master Thesis, 2020.
- [8] ATLAS Collaboration, «Search for an invisibly decaying Higgs boson or dark matter candidates produced in association with a Z boson in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector», *Phys. Lett. B* **776**, 318 (2018).
- [9] CMS Collaboration (A. Tumasyan *et al.*), «Identification of hadronic tau lepton decays using a deep neural network», *J. Instrum.* **17**, P07023 (2022).

- [10] ATLAS Collaboration, «Deep sets based neural networks for impact parameter flavour tagging in ATLAS», Technical report, ATL-PHYS-PUB-2020-014, 2020.
- [11] CMS Collaboration, «Performance of the DeepJet b tagging algorithm using 41.9 fb^{-1} of data from proton–proton collisions at 13 TeV with Phase 1 CMS detector», report number CMS-DP-2018-058; CERN-CMS-DP-2018-058.
- [12] P. Baldi, P. Sadowski, D. Whiteson, «Searching for exotic particles in high-energy physics with deep learning», *Nat. Commun.* **5**, 4308 (2014).
- [13] E. Arganda, A.D. Medina, A.D. Perez, A. Szykman, «Towards a method to anticipate dark matter signals with deep learning at the LHC», *SciPost Phys.* **12**, 063 (2022).
- [14] A. Celik, «A fast and time-efficient machine learning approach to dark matter searches in compressed mass scenario», *Eur. Phys. J. C* **83**, 1150 (2023).
- [15] A. Celik, «Exploring hidden signal: Fine-tuning ResNet-50 for dark matter detection», *Comput. Phys. Commun.* **305**, 109348 (2024).
- [16] C. Biscarat *et al.*, «Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC», *EPJ Web Conf.* **251**, 03047 (2021).
- [17] X. Ju *et al.*, «Performance of a geometric deep learning pipeline for HL-LHC particle tracking», *Eur. Phys. J. C* **81**, 876 (2021).
- [18] G. DeZoort *et al.*, «Charged particle tracking via edge-classifying interaction networks», *Comput. Softw. Big Sci.* **5**, 26 (2021).
- [19] F. Scarselli *et al.*, «The Graph Neural Network Model», *IEEE Trans. Neural Networks* **20**, 61 (2009).
- [20] DarkMachines community, «Unsupervised-Hackathon», July 2020, DOI: [10.5281/zenodo.3961917](https://doi.org/10.5281/zenodo.3961917).
- [21] T. Aarrestad *et al.*, «The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider», *SciPost Phys.* **12**, 043 (2022), [arXiv:2105.14027](https://arxiv.org/abs/2105.14027) [hep-ph].
- [22] The pandas development team, «pandas-dev/pandas: Pandas», Feb. 2020, DOI: [10.5281/zenodo.3509134](https://doi.org/10.5281/zenodo.3509134).
- [23] F. Pedregosa *et al.*, «Scikit-learn: Machine Learning in Python», *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [24] F. Chollet *et al.*, «Keras», <https://keras.io>, 2015.
- [25] T. O’Malley *et al.*, «Keras Tuner», <https://github.com/keras-team/keras-tuner>, 2019.
- [26] A. Hagberg, P.J. Swart, D.A. Schult, «Exploring network structure, dynamics, and function using NetworkX», Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM, USA, 2008.
- [27] C.R. Harris *et al.*, «Array programming with NumPy», *Nature* **585**, 357 (2020).

- [28] D. Grattarola, C. Alippi, «Graph Neural Networks in Tensor-Flow and Keras with Spektral». [arXiv:2006.12138](#) [cs.LG].
- [29] X. Li, L. Sun, M. Ling, Y. Peng, «A survey of graph neural network based recommendation in social networks», *Neurocomputing* **549**, 126441 (2023).
- [30] K. Han *et al.*, «Vision GNN: An Image is Worth Graph of Nodes», *Adv. Neural Inf. Process. Syst.* **35**, 8291 (2022). [arXiv:2206.00272](#) [cs.CV].
- [31] P. Pradhyumna, G.P. Shreya, Mohana, «Graph Neural Network (GNN) in Image and Video Understanding Using Deep Learning for Computer Vision Applications», in: «Proceedings of Second International Conference on Electronics and Sustainable Communication Systems (ICESC)», *Coimbatore, India, 4–6 August 2021*, pp. 1183–1189.
- [32] L. Wu *et al.*, «Graph neural networks for natural language processing: A survey», *Found. Trends Mach. Learn.* **16**, 119 (2023).
- [33] Y. Wang, Z. Li, A. Barati Farimani, «Graph neural networks for molecules», in: C. Qu, H. Liu (Eds.) «Machine Learning in Molecular Sciences», Challenges and Advances in Computational Chemistry and Physics, Vol. 36, *Springer, Cham 2023*, pp. 21–66.
- [34] P. Reiser *et al.*, «Graph neural networks for materials science and chemistry», *Commun. Mater.* **3**, 93 (2022).
- [35] J. Pata *et al.*, «MLPF: efficient machine-learned particle-flow reconstruction using graph neural networks», *Eur. Phys. J. C* **81**, 381 (2021).
- [36] F.A. Di Bello *et al.*, «Towards a computer vision particle flow», *Eur. Phys. J. C* **81**, 107 (2021).