

NAVE: Uma Solução Automatizada de Avaliação Educacional para Ambientes de Simulação de Redes de Computadores

Artur L. Melo
Instituto Federal do Rio Grande do
Norte (IFRN)
Macau, Brazil
m.artur@academico.ifrn.edu.br

Lucas A. Silva
Instituto Federal do Rio Grande do
Norte (IFRN)
Macau, Brazil
melo.coutinho@academico.ifrn.edu.br

Charles H. Santos
Instituto Federal do Rio Grande do
Norte (IFRN)
Macau, Brazil
charles.santos@ifrn.edu.br

Resumo

O NAVE (Network Analysis and Verification Engine) é uma ferramenta desenvolvida em Python que permite comparar configurações de roteadores em projetos do GNS3 com um modelo de referência. O sistema avalia as diferenças e gera uma pontuação de conformidade, facilitando a verificação e análise de topologias de rede simuladas. O projeto conta com interface gráfica, scripts de comparação automatizada e suporte a múltiplos projetos, servindo como um exemplo de aplicação prática de automação em redes de computadores. Ele foi desenvolvido a fim de automatizar a etapa de correção de atividades práticas realizadas em laboratórios de redes.

Palavras-chave

NAVE, Solução, Automação, Avaliação, Educação, Simulação, Emulação, Redes, Ferramenta, Correção

1 Introdução

A convergência dos serviços de telecomunicações para a Internet foi proporcionada por tecnologias como virtualização, orquestração de serviços, Redes Definidas por Software (SDN) e infraestruturas de computação em nuvem [8]. Visto que a Internet apresenta um crescimento acentuado, a implantação e a manutenção de novas infraestruturas dependem de profissionais especializados nas tecnologias mencionadas, entre outras. Relatórios indicam que a demanda por profissionais de TI, como engenheiros de redes, estão entre os que mais se expandem dentro do setor da tecnologia, crescendo 47% mais do que os demais empregos da economia [3].

Dessa forma, torna-se importante a formação desses profissionais a fim de suprir a demanda do mercado. Porém, devido ao alto nível de abstração em assuntos que envolvem redes de computadores, os estudantes frequentemente enfrentam dificuldades no processo de aprendizagem, ressaltando a importância das aulas práticas e de métodos avaliativos adequados [14]. Apesar de essenciais, as aulas práticas em laboratórios físicos nem sempre são viáveis, devido aos equipamentos possuírem rápida depreciação e necessitarem de manutenção periódica [7], ou pelo risco de comprometimento da rede institucional, caso não haja isolamento adequado [9].

Como alternativa, ferramentas de simulação de dispositivos de rede permitem a execução de experimentos sem a necessidade de uma infraestrutura física de laboratório. Entre essas ferramentas, destaca-se o GNS3 (Graphical Network Simulator 3), voltado para a emulação de ambientes de rede complexos. Originalmente criado para auxiliar na preparação para certificações Cisco, o GNS3 utiliza tecnologias como o Dynamips para emular o Cisco IOS, oferecendo suporte a imagens reais de roteadores e switches [2, 11].

Outra solução amplamente utilizada é o Cisco Packet Tracer, utilizado no ensino de redes de computadores por permitir a configuração de topologias, o teste de protocolos e a avaliação de comportamentos de rede, sem a necessidade de equipamentos reais [1]. Embora o Cisco Packet Tracer ofereça o Activity Wizard como mecanismo de correção automática, esse recurso é limitado a atividades desenvolvidas exclusivamente dentro do próprio ambiente simulado da Cisco [12].

Dessa forma, instrutores que não utilizam o Cisco Packet Tracer acabam realizando a correção dos exercícios manualmente [12], o que impõe uma maior carga de trabalho aos tutores durante o acompanhamento da evolução do aprendizado do aluno [10]. Para suprir essa limitação, ferramentas externas interagem com as plataformas de simulação e emulação para auxiliar nesse processo.

No entanto, ao melhor do nosso conhecimento, entre as soluções encontradas, existem limitações importantes dessas ferramentas no que diz respeito à ausência de uma camada visual de interação e à carência de um avanço na automatização desses recursos. Funcionalidades como uma interface gráfica, a exportação automática de notas para formatos de planilha e a correção em lote de várias topologias.

Nesse contexto, este artigo apresenta o NAVE (Network Analysis and Verification Engine), uma ferramenta desenvolvida para auxiliar instrutores de Redes de Computadores na correção automática de topologias e configurações de roteadores. O NAVE realiza comparações entre um projeto modelo e um ou mais projetos submetidos à avaliação, identificando discrepâncias e atribuindo pontuações configuráveis conforme o tipo de erro. Além disso, a ferramenta possui uma interface gráfica e a funcionalidade de exportação dos resultados das avaliações, permitindo integração com planilhas eletrônicas e potencializando melhorias no processo avaliativo.

O restante deste artigo está organizado da seguinte forma: a Seção 2 descreve os trabalhos relacionados ao tema de pesquisa, delineando as contribuições desta proposta; a Seção 3 apresenta a arquitetura do sistema NAVE, expondo seus principais componentes e funcionalidades; a Seção 4 apresenta a demonstração prática, ilustrando seu funcionamento, e a Seção 5 apresenta a conclusão do artigo.

2 Trabalhos Relacionados

De acordo com o contexto apresentado, o NAVE (Network Analysis and Verification Engine) é uma ferramenta voltada à verificação automática de projetos de redes simuladas no GNS3. A ferramenta permite comparar as configurações dos dispositivos com um modelo de referência, identificar erros de configuração e gerar uma pontuação de conformidade, auxiliando o processo de avaliação. Os

Tabela 1: Comparação simplificada entre o NAVE e trabalhos relacionados

Trabalho	Ambiente	Comparação	Gerar Pontuação	Identificar Erros	GUI	Multi-projetos
NAVE (este trabalho)	GNS3	Sim	Sim	Sim	Sim	Sim
Sarrió Ferrández [13]	GNS3	Parcial	Sim	Parcial	Não	Não
Emiliano [6]	GNS3	Não	Não	Não	Não	Não
Csengody et al. [4]	Packet Tracer	Sim	Sim	Sim	Sim	Não
Emiliano and Antunes [5]	GNS3	Não	Não	Não	Não	Não

trabalhos relacionados são analisados considerando essas funcionalidades.

Emiliano and Antunes [5] propõem uma extensão para o GNS3 capaz de gerar automaticamente as configurações dos roteadores Cisco presentes em uma topologia virtual. A solução lê o arquivo JSON gerado pelo GNS3 e extrai todas as informações sobre dispositivos e conexões. A partir desses dados, scripts em Perl interpretam a topologia, atribuem redes IP automaticamente, identificam interfaces e geram arquivos de configuração completos usando modelos pré-definidos. O resultado principal se caracterizou na redução da carga de trabalho durante o desenvolvimento de novos ambientes emulados.

Por sua vez, Sarrió Ferrández [13] desenvolve um conjunto de scripts para avaliar projetos de redes criados por alunos em simuladores. O objetivo é reduzir o tempo de correção, padronizar avaliações e permitir que os estudantes pratiquem configurações sem depender de laboratórios físicos. O autor constrói uma topologia completa de rede com sistemas de roteamento de redes, criando scripts em Python que verificam automaticamente a estrutura do projeto.

Uma aplicação capaz de gerar automaticamente configurações de roteadores a partir de projetos criados no GNS3 é proposto por Emiliano [6]. O projeto utiliza arquivos em formato JSON exportados pelo simulador para interpretar a topologia e criar scripts de configuração personalizados para cada topologia criada, trabalhando como a criação de scripts manuais de configuração dos dispositivos, que devem ser aplicados pelo usuário.

Csengody et al. [4] apresentam uma ferramenta para automatizar a avaliação das configurações feitas por alunos em dispositivos Cisco. A solução utiliza um modelo chave definido pelo professor como referência para os exercícios. As configurações produzidas pelos alunos são importadas, convertidas e comparadas ao modelo.

O NAVE se diferencia ao integrar, em uma única ferramenta, a análise completa das configurações, a verificação da topologia, a geração automática de pontuação, uma interface gráfica e a exportação de resultados para planilhas, oferecendo um fluxo mais eficiente e adequado às atividades práticas.

3 Arquitetura

O NAVE é uma ferramenta de apoio destinada a instrutores de redes de computadores, com o objetivo de automatizar a correção de atividades práticas realizadas em laboratórios que utilizam o ambiente de emulação GNS3. De modo geral, seu funcionamento consiste em comparar topologias definidas como projetos do GNS3. Em todos os casos, um modelo é utilizado como referência para identificar as discrepâncias em relação a um projeto enviado pelo

aluno. Durante esse processo, é registrado um relatório estruturado contendo as diferenças encontradas e a pontuação atribuída a cada topologia analisada.

Para este objetivo, uma arquitetura funcional foi projetada para compor um conjunto de componentes e funcionalidades que se integram para permitir a análise, visualização e comparação de topologias de rede. Conforme ilustrado na Figura 1, os componentes principais se organizam nos módulos **User Interface**, que incluem as funcionalidades da interface do usuário, e **Core Services**, que contêm os componentes e módulos necessários para as correções automatizadas.

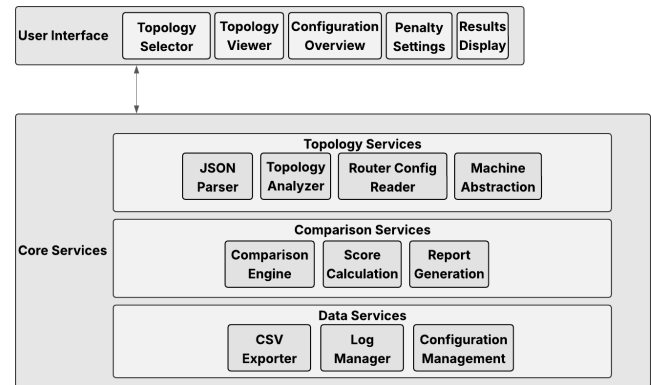


Figura 1: Arquitetura do NAVE

A arquitetura do **Core Services** consiste em três componentes. A **Topology Services** realiza a leitura, interpretação e organização das topologias selecionadas pelo usuário, convertendo os dados brutos do GNS3 em classes estruturadas que serão utilizadas na comparação. O **Comparison Services** constitui o núcleo lógico da ferramenta, executando a análise entre a topologia de referência e as topologias avaliadas, registrando divergências e produzindo informações que fundamentam tanto os relatórios quanto a pontuação final. Por fim, o **Data Services** oferece suporte ao armazenamento das penalidades, à exportação dos resultados de comparações em lote e à geração de arquivos .log a partir dos relatórios emitidos pelo módulo de comparação.

3.1 User Interface

O módulo da interface gráfica da ferramenta NAVE foi projetada para fornecer aos instrutores acesso aos seus mecanismos de análise e correção de topologias. A interface gráfica oferece recursos que

permitem a comparação, visualização e avaliação de topologias de rede desenvolvidas no GNS3. A seguir, descrevem-se as principais funcionalidades disponibilizadas.

- **Topology Selector:** Esta funcionalidade tem como principal função a coleta e seleção das topologias de referência (criada pelo instrutor) e da topologia a ser comparada (criada pelo aluno). É possível selecionar este de forma individual, sendo também possível selecionar vários para verificações em lote. A realização das comparações é feita pelo componente **Topology Services**, cujos resultados são disponibilizados pelo **Results Display**.
- **Topology Viewer:** Permite que uma pré-visualização dos projetos selecionados seja gerada na tela, desenhando em um quadro os dispositivos em suas posições lógicas, seus identificadores e suas ligações a outros elementos.
- **Configuration Overview:** Esta função apresenta uma visão detalhada da topologia e das configurações dos roteadores. São exibidas informações como nome, tipo, identificador (ID), posição dos nós e conexões entre eles. Além disso, a ferramenta gera um resumo das configurações de rede, incluindo interfaces, endereços IP e rotas configuradas em cada dispositivo.
- **Penalty Settings:** Tem como função exibir e permitir a alteração das penalidades utilizadas no cálculo da pontuação final. A coleta e a atribuição dessas penalidades são feitas pelo **Configuration Management**. As penalizações disponíveis abrangem: *roteador inexistente*, *diferença de interface*, *diferença de endereço IP* e *diferença de rota*.
- **Results Display:** É responsável por exibir ao usuário os resultados obtidos após a execução de comparações ou análises realizadas pelos **Topology Services**. Sua principal função é apresentar, de forma clara e interativa, as informações geradas pelos módulos de verificação e avaliação, permitindo uma visualização imediata dos resultados.

3.2 Topology Services

O Componente *Topology Services* é responsável por centralizar e gerenciar as operações relacionadas às topologias de rede processadas pela ferramenta. Sua função principal consiste na análise e interpretação das topologias, a fim de extrair informações que serão usadas posteriormente pelos **Comparison Services**.

- **JSON Parser:** Os projetos criados pelo *GNS3* são, essencialmente, arquivos no formato JSON. Com base nisso, o JSON Parser tem como principal função a extração das informações referentes à topologia. Esses arquivos contêm dados sobre os identificadores, tipos, posições e conexões dos nós da rede.
- **Topology Analyzer:** Esta funcionalidade é responsável por localizar e classificar as informações obtidas a partir da leitura em formato JSON dos projetos gerados pelo *GNS3*. Sua principal função consiste em interpretar os dados brutos da topologia de rede e fornecê-los ao **Router Config Reader** e **Machine Abstraction**.

- **Router Config Reader:** O sistema identifica os dispositivos do tipo roteador e busca, para cada um deles, o arquivo de configuração correspondente, nomeado no formato `<identificador>_startup-config.cfg`. Quando encontrado, o arquivo é processado para extrair as configurações relevantes, que são armazenadas em estruturas organizadas para a abstração posterior pelo **Machine Abstraction**. Entre os parâmetros extraídos estão as interfaces presentes no roteador, seus endereços IP e as rotas configuradas.
- **Machine Abstraction:** Tem como função abstrair os elementos da topologia e as configurações encontradas em suas devidas estruturas, a fim de simplificar a complexidade de uma rede de computadores. O objetivo dessa camada de abstração é transformar os elementos brutos de uma topologia em objetos estruturados, facilitando a comparação automatizada das topologias pelo **Comparison Services**.

3.3 Comparison Services

O componente *Comparison Services* é responsável pela análise comparativa entre as topologias selecionadas e as configurações dos roteadores obtidas pelos **Topology Services**. Seu objetivo é identificar diferenças estruturais e de configuração, atribuindo uma pontuação com base em critérios de correção previamente definidos pelo usuário.

- **Comparison Engine:** A funcionalidade Comparison Engine é responsável por comparar os objetos estruturados da topologia de referência com as topologias geradas pelos alunos, tendo como critérios a ausência de interfaces, dispositivos, diferença de endereços IPs e conexões entre os elementos. Esses critérios são então utilizados posteriormente para o cálculo da nota pelo **Score calculation**.
- **Score Calculation:** Essa função calcula a pontuação final de uma comparação entre uma topologia referência e a topologia gerada pelo aluno, utilizando o relatório gerado pelo **Report Generation** como base e reduzindo uma pontuação previamente definida pelo usuário para cada inconsistência encontrada.
- **Report Generation:** É responsável pela criação automática de relatórios após a execução do processo de comparação entre topologias e configurações de roteadores pelo **Comparison Engine**. Essa funcionalidade visa registrar de forma sistemática todos os resultados da análise, incluindo as diferenças identificadas e a pontuação final obtida pelo **Score Calculation**.

3.4 Data Services

O componente *Data Services* é responsável pela gestão e manipulação dos dados produzidos e utilizados pela ferramenta. Ele atua como uma camada de suporte para a leitura, gravação e exportação das informações geradas durante o processo de análise e comparação de topologias.

- **CSV Exporter:** A funcionalidade *CSV Exporter* tem como função principal converter os arquivos de log em planilhas no formato `.csv`, atribuindo o nome do projeto do aluno que foi comparado e sua nota final. Esse processo facilita a

análise dos resultados e a integração com outras ferramentas de visualização.

- **Log Manager:** O *Log Manager* é responsável pela criação dos arquivos de registro (.log) que tem como base os relatórios gerados pelo **Report Generation**. Esses registros contêm informações sobre erros, alertas, comparações de interfaces e rotas, além da pontuação final obtida.
- **Configuration Management:** Tem como principal função a leitura e modificação das penalidades salvas anteriormente pelo **Penalty Settings**, que estão armazenadas em um arquivo .cfg, a fim de persistir as preferências do avaliador entre as sessões.

3.5 Metodologia de desenvolvimento

Para o desenvolvimento do NAVE, foi adotada uma metodologia incremental, na qual cada módulo foi implementado de forma progressiva até a consolidação do sistema completo. Inicialmente, foram desenvolvidos os módulos responsáveis pela leitura e interpretação dos projetos do GNS3. Em seguida, os módulos de comparação das topologias e configurações foram implementados, e por fim, a interface visual da ferramenta.

Adicionalmente, foram utilizados modelos de inteligência artificial generativa para auxiliar na geração, refinamento e resolução de erros no código. Isso foi feito de tal modo a acelerar o desenvolvimento e a implementação dos componentes funcionais da solução, principalmente nas etapas de manipulação de dados e interface gráfica.

4 Demonstração

Nesta seção, apresenta-se a demonstração prática do NAVE, evidenciando seu fluxo de utilização, desde a seleção das topologias até a geração do relatório final. Para esta demonstração, considera-se uma comparação simples em relação ao projeto de referência. Além disso, um **vídeo demonstrativo do NAVE** está disponível em plataforma online.

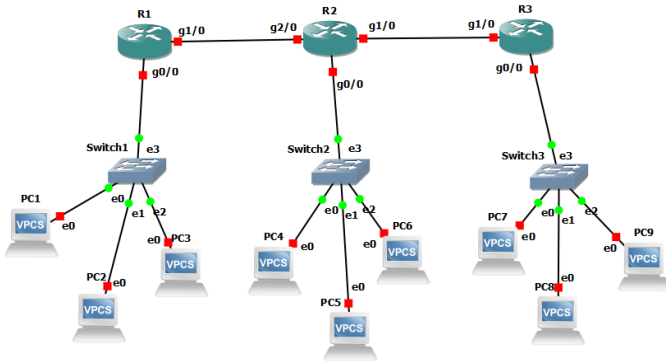


Figura 2: Topologia modelo

Para a demonstração do fluxo de funcionamento, utilizou-se uma rede de referência, conforme ilustrado na Figura 2. A topologia com falhas apresenta a mesma estrutura da topologia modelo. Entretanto, as configurações dos roteadores não foram devidamente ajustadas, o que impede seu correto funcionamento. É importante destacar

que a leitura das configurações só é possível quando esses arquivos são previamente extraídos do próprio GNS3, utilizando a função **Export config**, ao clicar com o botão direito sobre um roteador. O usuário deve garantir que os arquivos de configuração gerados estejam armazenados em uma pasta denominada **dynamips**.



Figura 3: Página inicial do NAVE

Para iniciar o processo de comparação, o usuário deve abrir o arquivo `gui.py`, o que exibirá a página inicial da ferramenta (Figura 3). Em seguida, na barra de navegação superior, seleciona-se a opção **Comparar projeto**, onde é possível escolher tanto a topologia modelo quanto a topologia a ser avaliada. Após a seleção, a ferramenta apresenta uma pré-visualização da topologia na área inferior da interface, auxiliando na identificação do projeto carregado.

Após a seleção das topologias, basta acionar o botão **Comparar** para iniciar o processo de avaliação. Em seguida, a ferramenta gera um relatório que é exibido em uma nova janela, apresentando cada discrepância identificada, o valor da penalidade associado a cada uma delas e, por fim, a pontuação final da topologia avaliada. Além da exibição na interface, o relatório também é automaticamente salvo em um arquivo .log, contendo o nome do projeto, a data e o horário em que o processo foi executado.

O processo de comparação em lote segue a mesma lógica da avaliação individual, porém, permite analisar diversas topologias simultaneamente. Para utilizá-lo, o usuário deve selecionar a opção **Comparar Lote de Projetos** na barra de navegação superior e, em seguida, escolher a topologia modelo e a pasta que contém as topologias dos projetos a serem avaliados. Após a seleção da pasta das topologias, a ferramenta identifica os projetos e exibe uma lista com todas as topologias encontradas.

Para alterar os valores das penalidades, o usuário deve clicar no botão **Penalizações**, que exibirá os valores atuais e permitirá a sua modificação. Após realizar os ajustes desejados, é necessário acionar o botão **Salvar** para que as novas penalidades sejam aplicadas e persistidas no sistema.

Além dessas funcionalidades, a ferramenta também possibilita a visualização de todas as configurações encontradas em um projeto, conforme ilustrado na Figura 4. Para isso, basta utilizar a opção

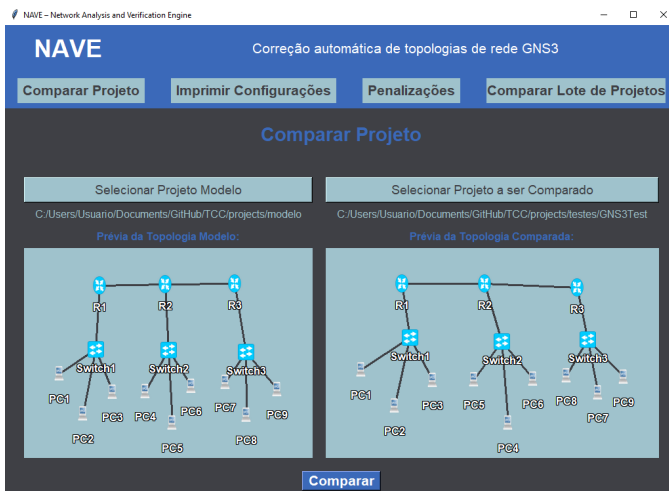


Figura 4: Tela de comparação individual

Imprimir Configurações e selecionar o projeto cuja configuração se deseja consultar.

5 Conclusão

Este trabalho apresentou o NAVE (Network Analysis and Verification Engine), uma ferramenta para auxiliar instrutores de Redes de Computadores ao oferecer correção automática de topologias GNS3 e de configurações de roteadores. Para isso, o sistema utiliza um projeto a ser usado como modelo definido pelo instrutor e um ou mais projetos submetidos pelos estudantes. Além disso, a ferramenta disponibiliza uma interface gráfica intuitiva e suporte para exportação dos resultados em formato de planilha.

Com o NAVE, o processo avaliativo de estudantes de Redes de Computadores pode se tornar mais rápido e eficiente, reduzindo o tempo necessário para a etapa de correção de atividades práticas. Embora a ferramenta, na sua versão atual, esteja limitada ao ecossistema do GNS3 e à análise de configurações estáticas, ela estabelece uma base sólida para futuras extensões que ampliem seu escopo e funcionalidade. Dessa forma, o NAVE contribui para aprimorar a formação de profissionais de redes, apoiando a preparação de estudantes em um cenário marcado pela crescente demanda por especialistas na área.

Referências

- [1] Jordan Allison. 2022. Simulation-based learning via cisco packet tracer to enhance the teaching of computer networks. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol. 1*. 68–74.
- [2] DP Sumit Kumar Ansari, Mishra. 2018. A Study and Designing of Custom Network Configuration using Integrated Protocols in SDN Architecture. *International Journal of Trend in Scientific Research and Development 2* (2018). Issue 4.
- [3] CompTIA. 2024. State of the Tech Workforce 2024. <https://www.comptia.org/en-em/resources/research/state-of-the-tech-workforce-2024>. Relatório de mercado de trabalho em TI.
- [4] Z Csengody, D Macko, and K Jelemenska. 2018. Automated Evaluation of a Network Device Configuration. In *2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, 99–104.
- [5] Rodrigo Emiliano and Mário Antunes. 2015. Automatic network configuration in virtualized environment using GNS3. In *2015 10th International Conference on Computer Science & Education (ICCSE)*. IEEE, 25–30.
- [6] Rodrigo Jose Lopes Emiliano. 2015. *Gerador Automático de Configurações de Equipamentos de Rede Em Ambientes Virtualizados*. Master's thesis. Instituto

Politecnico de Leiria (Portugal).

- [7] Rafael de Magalhães Dias Frinhani, Adriana Prest Mattedi, and Rarish Costa Souza. 2023. Um estudo comparativo entre modelos de utilização de equipamentos de tecnologia da informação para práticas em laboratórios de computação. (2023).
- [8] Global Growth Insights. 2024. Network Infrastructure Market Size, Share, Growth 2024–2033. <https://www.globalgrowthinsights.com/market-reports/network-infrastructure-market-116404>. Acesso em: 28 nov. 2025.
- [9] John MD Hill, Curtis A Carver Jr, Jeffrey W Humphries, and Udo W Pooch. 2001. Using an isolated network laboratory to teach advanced networks and security. *ACM SIGCse Bulletin* 33, 1 (2001), 36–40.
- [10] Juanita J Hutchison. 2022. *The Impact of Instructor Workload on the Use of Written Corrective Feedback*. Ph.D. Dissertation. Walden University.
- [11] Jason C Neumann. 2015. *The book of GNS3: build virtual network labs using Cisco, Juniper, and more*. No Starch Press.
- [12] Neville Palmer. 2019. An application to automate assessment in a computer networking laboratory. In *2019 25th International Conference on Automation and Computing (ICAC)*. IEEE, 1–6.
- [13] Marco Sarrió Ferrández. 2023. GNS3-network simulator for evaluation of network projects using scripts. (2023).
- [14] Kai Zhao. 2024. Impact of Laboratory Practice on Computer Network Teaching. *Advances in Computer, Signals and Systems* 8, 2 (2024).